Technische Universität Dresden
Fakultät Informatik

Diplomarbeit

A Camera Sensor Model for Sensor Fusion

Matthias Fichtner

Supervisor: Prof. Dr. Steffen Hölldobler,
Dr. Axel Großmann

18th October 2002

Abstract

The ability to navigate robustly is of high importance for autonomous, mobile robots. The agent's decisions are based on the internal representation of its environment, which strongly depends on the appropriate interpretation of sensory information. Naturally, perception in a real-world environment is affected by errors, noise, sensor unreliability, ignorance, etc. The combination of information from several, perhaps different sensors of various characteristics and their integration into a shared representation taking uncertainty into account is the field of sensor fusion. For that, a sensor model accomplishes data alignment to yield quantities of the same type as other sensor models. In this report, I describe a sensor model for a CCD camera mounted on a mobile robot, that is suitable for sensor fusion. Being based on visual features, the agent first extracts line segments from an image and simulates world model projection. Then, the match of both feature maps is evaluated using a similarity measure, where I have devised several measures. The sensor model was obtained from comprehensive stochastic sampling and exhibited very promising results in computer simulations as well as real-world experiments.

Contents

List of Figures

Chapter 1

Introduction

In order to successfully operate in real-world environments in a goal-directed manner, an autonomous agent needs to get to know the current state of the environment and to maintain information about the past. The agent starts with some provided a priori information, typically a geometric or topological map of the environment. During operation, the agent continuously updates its internal representation of the environment including the map, which is called the *world model*. Clearly, the only means an agent has to update the world model, is to measure certain aspects of the environment in its immediate vicinity with the available sensors. In real-world applications like robotics, the perceptive capabilities are often very limited by construction, i.e., by the number, type, range, quality, configuration, etc. of the sensors our agent is employed with. Hence, it is necessary to equip the agent with those sensors that enable it to perceive significant aspects of the environment with respect to the tasks to solve in a reliable and sufficient manner.

Moreover, percepts are intrinsically uncertain. There are various aspects of uncertain knowledge, making it difficult to define uncertainty and to distinguish between these aspects: incompleteness, inaccuracy, imprecision, unreliability, and erroneous data.

Further aspects of uncertainty are vagueness, variability and ambiguity.

It is important to recognise what the actual sources of uncertainty are — they are to be seen in the perception and interpretation process of the agent itself. From that, the vital requirement arises that an autonomous agent perceiving the real-world environment is aware of its own uncertainties, i.e., taking it into account in information interpretation.

Looking at humans and animals in particular, moving between different places plays an important role in their life. A good definition for *navigation* from the cognitive perspective is: "Survival of the individual is contingent upon adaptive skills to find, learn and return to specific places (and often at specific times) — such as the nest or the feeding site — quickly and safely." [Trullier et al., 1997, p. 1] Navigation is of similar importance for mobile robots, especially if autonomy is desired. But still, their navigation methods are specialised and brittle [Meyer and Guillot, 1994].

So far, research has established some prominent methodologies for reasoning and acting with uncertainty. In general, they concern different aspects of uncertainty, like unreliability and ignorance. Often, a method is only appropriate for one or a few particular concepts of uncertainty, mostly unreliability. Therefore, when designing an intelligent agent, it is important to check the suitability of different methodologies; or to cite Krause and Clark [1993, p. 101], "It is really a matter of ensuring that the model fits the application domain."

An interesting approach to handling and interpreting uncertain information is *sensor fusion*. It aims at decreasing the uncertainty inherent in sensory information by integrating information from several, preferably different sensors into a common world representation. In contrast to many ad hoc approaches designed for specific sensor types and uncertainty concepts, sensor fusion is to be seen as a generic methodology. It exploits the redundancy of available information sources and can gain increased semantic knowledge than would be possible considering either alone [Kortenkamp and Weymouth, 1994, p. 4]. When several sources of sensory information are available, sensor fusion accomplishes data integration for redundant parts and data aggregation for the remaining parts. Concerning data integration, sensor fusion allows to recognise inconsistencies and errors from conflicting information, or sensor unreliability and incompleteness from redundancy, for instance. In summary, different advantageous sensor characteristics can be combined and mistaken behaviour recognised.

The project at hand addresses sensor fusion for an autonomous, mobile robot. Before perceptive information from different sensor types can be integrated by sensor fusion into a common referential, the sensor-specific information needs to be transformed into an independent description. This conversion is defined by a sensor model.

The development of a camera sensor model to be applied to our mobile robot in an office environment is the integral content of this thesis. First, a three-dimensional, geometric world model of our office environment has been built (by hand) comprising line segments that are easily discernible. Based on this world model, line features are extracted from the camera image and the world model. Then, both feature maps are compared employing a appropriate similarity measure. In this project, several similarity measures have been devised and evaluated in experiments. The best one was chosen to be employed in the camera sensor model. In contrast to the classical approach to visual navigation that involves a computational expensive search for correspondences among features of both feature maps, the similarity measure proposed here is based on a very simple and fast matching technique correspondence establishment (cf. Lowe [1987], Lanser [1997], Kumar and Hanson [1994]). Finally, the sensor model was computed in comprehensive computer simulations by stochastic sampling and abstraction to pose deviations.

The outline of this report is as follows: Chapter 2 describes the aim of sensor fusion and prominent approaches thereof. Chapter 3 is devoted to previous work in the topics involved in devising sensor models in general and a model for visual sensors in particular. Then, Chapter 4 comprises the detailed explanation of this project addressing the implementation of a camera sensor model for sensor fusion. Correspondingly, Chapter 5 is devoted to the experiments performed employing the devised camera sensor model. Besides empirical results obtained from both, computer simulations and real-world experiments, a comprehensive analysis and discussion on the results are included. Finally, Chapter 6 provides a general conclusion on the implementation and empirical results. Based on the findings, it further comprises suggestions on improvements and future work.

Chapter 2

Sensor Fusion

The general methodology of integrating information from several, possibly different sensor devices into a common, unique representation is called *sensor fusion*. Its aim is to gain improved accuracy or higher certainty of the perceived information by exploiting the synergetic effect of combining independent sources of information than would be possible considering them separately (see Section 2.3 for definitions). Furthermore, it eases the task of information processing and retrieval, since first, sensor fusion takes care of appropriately integrating and representing information from various sensory sources, and second, it maintains a unique representation structure.

Roughly speaking, the integration of perceptive information by sensor fusion comprises the following stages:

1. First, percepts are acquired from sensors.
2. The data often has to be pre-processed.
3. Its integration into the common representation requires to adopt the percepts to the target data structure.
4. Correspondences between new percepts and entities already known are to be established.
5. Depending on the correspondences, a percept is either added to the common representation or the corresponding entry is updated according the policy in use.

Since steps 1–3 are specific to the sensor in use, they are summarised in the so-called *sensor model* for this sensor device. Step 5 is the most important stage in sensor fusion and may comprise rather complex reasoning, e.g., if conflicting or erroneous information was recognised. Besides these stages regarding data integration, sensor fusion also comprises concepts to reason about the sensory data, for instance in order to adopt to changing conditions. Finally, the acquired and maintained knowledge is presented for usage in different applications.

In contrast to sensor fusion, ad hoc approaches implement sensor- and task-specific methods to access, represent and interpret percepts in a non-modular manner, which makes it difficult to substitute a sensor or add another one.

The approach of sensor fusion is not recent, but only recently principles were established and formalisms defined. While the general term for combining information from various sources is often called "data fusion" or "information fusion", several branches evolved according to different application domains such as sensor fusion.

The following definition for data fusion stems from the military defence community, where data fusion in general is an interesting topic as well:

Data fusion is a "multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources". [DSTO, 1994]

2.1 Motivation

Emphasising that every technical sensor has a limited precision, is subject to thermal noise and may fail at all, the advantages of sensor fusion are in particular:

- Exploiting the redundancy when several, statistically independent sensors of the same (or similar) type are in use, sensor fusion reduces the vulnerability of the whole system to failures of a single device.
- Integrating measurements from the same sensor at different times reduces the effect of noise, since measurement values are correlated, but noise is independent.
- Combining measurements from different sensor devices perceiving the same environment often yields higher accuracy and reliability, if advantageous characteristics of several sensors are appropriately integrated. More importantly, failures of one sensor type caused by illusions that produce conflicting data during the fusion process can be recognised and resolved, e.g., by controlling the focus among the information sources. By that, sensor fusion defines a method to combine measurements and to draw inferences about their evidential contributions. Employing several different sensor devices also reduces ambiguity in data caused by multiple hypotheses.

A further interesting property of sensor fusion is gained if a single, highly sophisticated and expensive sensor device can be replaced with a suite of simple sensors that achieves the same confidence level but at considerable less costs. For instance, in [Fennelly et al., 1993] a system of six X-ray explosion detectors achieves the same 0.95 confidence level as its substitute, while each sensor taken separately only has a confidence level of 0.50. The whole system was of much lower price and is unlikely to fail completely at the same time.

A significant proportion of inspirational findings comes from biological and cognitive studies, whereof a few concepts are described subsequently.

2.2 Cognitive Inspiration

An impressive example for successful sensor fusion is the human and animal brain, which continuously integrates information from all senses in order to update our representation of the environment, to adapt our beliefs, to draw better inferences and to improve accuracy of information, for instance.

A comprehensive definition for *sensor fusion* concerning uncertainties in particular is given by Murphy [1996, p. 2]: "Intelligent sensor fusion is defined as a process which can autonomously gather observations from multiple sensors and combine them into a single, coherent percept (execution), adapt the combination process to major environmental changes and sensor malfunctions (exception handling), and can determine its own sensing strategies for observing the percept (configuration)."

According to Bower [1974], a perceptive process performs in one of four levels, which is determined by the expected degree of discordance between information from different sensors and is fixed for the specific task by evolution:

1. "Complete sensory unity": ignoring discordance; no effort is made towards recognition of conflicting information.
2. "Unity with awareness of discordance and the possibility of recalibration": conflicting information is recognised and the integration process of the offending sensors is recalibrated.
3. "Unity with awareness of discordance and tendency toward suppression": no recalibration is performed, instead the integration of information from the offending sensors is suppressed.
4. "No unity at all": the sensory information comprise no commensurable attributes, thus cannot be associated [Bower, 1974].

Each perceptive mode describes the evidential contribution by the sensors. In traditional robotics, different modes of information fusing are hardly considered. Hence, for instance faulty sensors cannot be recognised and sensory illusions are ignored.

Furthermore, different sensors naturally contribute differently to the fusion process in accordance to their perceptive capabilities, e.g., vision is the most influential sense for humans.

The psychophysical studies of Marks [1978] showed that there exists a set of common attributes to all human senses. Based on that, the literature suggests that after a sensor-specific pre-processing was applied, all information is integrated into a shared representation.

Subsequently, some terms are defined, that will be used throughout the report.

2.3 Classifications

Uncertainty being a very vague term with many flavours, it is important to distinguish between some aspects of *uncertainty* in information as follows:

Incompleteness means lack of information, be it missing specific samples of a data series or a complete modality of sensory information according to the agent's equipment. Besides signal-level (raw) percepts, incompleteness also concerns limited derived knowledge, e.g., due to non-decidability or resource constraints, including loss of information or semantics during data processing.

Ignorance resembles the concept of incompleteness, but emphasises the state of knowing about the absence of information.

Accuracy designates the suitability of a method or model [Lamb et al., 1994]. This depends on the methods used during processing, manipulation and representation of information among others.

Imprecision concerns the "degree of detail recorded or encoded in the measurements" [Lamb et al., 1994]. Imprecision can be caused by insufficient means of internal representation (data structure, resolution), and by data manipulation, analysis or generalisation that accumulate imprecision.

Unreliability exhibits the degree of confidence in the measurement. Possible sources degrading the confidence are technical robustness of sensor devices, and environmental influences on perception and transmission of data, e.g., noise.

Errors may always be present in measurements, e.g., due to broken sensors or (human) carelessness during the data accumulation or manipulation process. Since errors can affect data significantly, they may mislead the agent's decisions that are based on those data. The recognition of errors is sometimes difficult and involves reasoning about past information.

The sensor fusion process usually comprises three stages:

Correlation: also called *data alignment*, converts the sensor-specific attributes into a commensurable representation. If no alignment technique exists, the information from this sensor cannot be integrated into the shared representation at hand. The alignment of percepts is accomplished by a specific sensor model for each sensor type.

Association: In this stage, relationships between entities of sensory data obtained from different sources are established, since the perceived attributes of the same object in the world should refer to the same, unique object of the internal representation. By that, a set of tracks is obtained for each object, like the robot's path, velocity etc.

Combination: describes the aggregation of values of all commensurable attributes and all information sources that refer to the same target object into a unique, updated value.

While these stages are performed during sensor fusion independent to the type of percepts considered, different kinds of sensory information are distinguished with respect to their semantic level of abstraction, similarly to other information processing techniques:

Raw measurements: Fusion of several signal-level items reduces variance in measurements. Operating with raw measurements usually infers the need to process a high amount of data. No abstraction from specific conditions is achieved at this level. Pre-processing usually does not increase its semantic level.

Attributes: The extraction of features from measurements yields a semantic interpretation of the data. The result is a meaningful description concerning the properties under consideration. By that, a certain level of abstraction from specific conditions to a more general description is gained, which often is part of the data alignment stage of sensor fusion. Within this abstraction level there may be several sub-levels, e.g., composite features.

Decisions: On an even higher semantic level, the symbolic level, fusion of decisions or rules is performed. Here, the aim is to reconcile the current decisions (beliefs) on arrival of new interpretations of percepts, i.e., decisions (inputs) are fused in order to yield new inferences.

If the perceptive inputs at signal level from all sources have the same, homogeneous quantity, they can be fused without prior manipulations. Otherwise, sensor fusion is only possible on a higher abstraction level.

2.4 Approaches

A few prominent approaches to sensor fusion are described in this section each taking the challenge of implementing certain aspects of uncertainty: Bayesian probability theory on one hand, and Dempster-Shafer theory and Fuzzy logic on the other hand, the latter being the two main alternatives to probability theory since the 1970s.

Although the Certainty Factor Model (Shortliffe [1976], Buchanan and Shortliffe [1984]) belongs to the set of prominent approaches, it is nowadays "of historical interest only" and therefore left out here [Russell and Norvig, 1995, p. 467].

Early implementations of sensor fusion in robotics were ad hoc approaches and go back to the 1980s (cf. Herman and Kanade [1986], Brooks [1985], Crowley [1985], Chatila and Laumond [1985] to name a few). Later, Durrant-Whyte [1987] published theoretical techniques for manipulating uncertainties in perception based on estimation theory. In general, most techniques developed for robotic sensor fusion apply and extend methods already known in control and estimation theory for a while; these include the Kalman filter (Kalman [1960], Kalman and Bucy [1961]), the Extended Kalman Filter [Jazwinski, 1973] and Monte Carlo methods (Handschin [1970], Handschin and Mayne [1969], Gordon et al. [1993]) like Sampling Importance Resampling (SIR) [Rubin, 1988] or Rejection Sampling [Hammersley and Handscomb, 1964].

2.4.1 Bayesian Probability Theory

A simple form to integrate two evidences, m_1, m_2, using Bayesian probability theory into the state description (vector) x, is defined by

$$P(x|m_1, m_2) \propto P(x)P(m_1|x)P(m_2|x) \tag{2.1}$$

which is based on Bayes' rule $p(x|m) = p(m|x)p(m)/p(x)$ [Russell and Norvig, 1995]. Here, $P(x)$ denotes the joint probability distribution of the state vector x. The state description x comprises a set of *state variables* which represent significant properties of the environment and is measured by sensors.

Formula 2.1 means that the posterior probability density for the state vector x is proportional to the prior of x multiplied by the probability for the evidences m_1 and m_2. Note that the evidences are assumed to be independent to each other, i.e., $P(m_1|x, m_2) = P(m_1|x)$ and $P(m_2|x, m_1) = P(m_2|x)$, which substantially reduces the number of conditional probabilities. Apparently, the evidences measure the same numerical quantity, i.e., neither data alignment nor correspondence establishment is required since they are directly commensurable.

According to Bayesian probability theory, estimating the state x_t of the system at time t given measurements $\mathbf{m}_t = m_1, \ldots, m_t$ is obtained by computing the probability density function (PDF) $p(x_t|\mathbf{m}_t)$. Assuming that measurements up to time $\tau \leq t$ are independent given the state vectors $\mathbf{x}_t = x_1, \ldots x_t$, the PDF can be obtained using factorisation applied to Bayes' rule [Fearnhead, 1998, p. 10]:

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{m}_\tau) &\propto p(\mathbf{x}_t) \prod_i^\tau p(m_i|\mathbf{x}_t) \\ &\propto p(\mathbf{x}_t) \prod_i^\tau p(m_i|x_t) \end{aligned}$$

In order to obtain the desired state description $p(x_t|m_\tau)$, the marginal distributions for $x_1, \ldots, x_{t-1}$ have to be formed. Since this and computing the normalisation factor involves high-dimensional integration, the following incremental computation scheme is preferred in general [Fearnhead, 1998, p. 11]:

$$p(x_t|\mathbf{m}_{t-1}) = \int p(x_{t-1}|\mathbf{m}_{t-1})p(x_t|x_{t-1})\mathrm{d}x_{t-1} \tag{2.2}$$

$$p(x_t|\mathbf{m}_t) \propto p(m_t|x_t)p(x_t|\mathbf{m}_{t-1}) \tag{2.3}$$

While Equation 2.2 expands the sequential *Markov chain* [Papoulis, 1984, p. 532] to the next time instance, which is called *prediction step*, Formula 2.3 incorporates the new measurement into the PDF bearing the name *update step*. For short, a Markov chain describes a sequence of state values, x, where each is completely determined by the previous one in accordance to $p(x_t|x_{t-1})$.

Likewise as above, all percepts measure the same numerical quantity that is fully commensurable with the state description, such that data alignment and correspondence establishment are implicitly assumed. Data alignment accomplishes a certain independence with respect to the sensors and data abstraction in use.

The Kalman filter is derived from the general Bayesian equations given above by introducing the assumptions of each state variable being a Gaussian PDF, the transition model being linear, and the uncorrelated noise being Gaussian as well (Kalman [1960], Kalman and Bucy [1961], Sorenson [1970]). This simplifies the involved prior and posterior distributions to be Gaussians as well, making the inversion from $p(m_t|x_t)$ to $p(x_t|m_t)$ easy. It has become a very prominent technique in various domains with linear state transition systems (cf. Lowe [1989], Liang et al. [1991], Azuma [1995], Friedmann et al. [1992], Watson [1986]).

Considering the application of Bayesian state estimation to an autonomous agent, the typical perceive-decide-act sequence of each step is modelled as illustrated in Figure 2.1. The agent obtains percepts o from its environment by

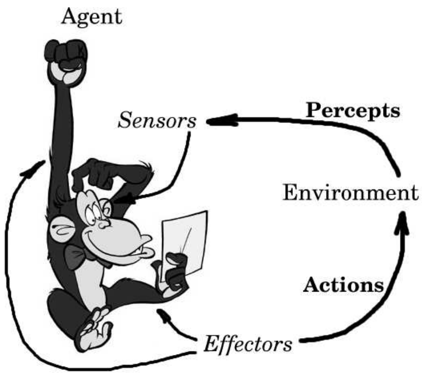


Figure 2.1: Interaction model between agent and world.

means of its sensors, and can perform actions a in the environment using its effectors. The decision which action to take is based on the agent's current belief x how the world is like (the world representation), which in turn is based on previous percepts and actions.

A suitable mathematical model for this interaction process is called Markov Decision Problem (MDP) (cf. Kaelbling et al. [1998]) and is described by the tuple $\langle X, A, T, R \rangle$, where

- X is a finite set of states of the environment;
- A denotes a finite set of actions the agent can perform;
- T denotes the state transition function $T : X \times A \rightarrow \Pi(X)$ which defines a probability distribution of ending in a certain state $X = x$ when the agent performs an action in a certain state;
- R defines a reward function $R : X \times A \rightarrow \mathbb{R}$ that determines the expected immediate reward which the agent gains for taking a certain action in a certain state, only mentioned for the sake of completeness.

In the Bayesian framework, the value of the world state representation x_t is denoted (see also Section 3.3):

$$Bel(x_t) = P(x_t|\mathbf{x}_{t-1}, o_1 \dots o_{t-1}, a_1 \dots a_{t-1})$$

which means that the *belief* at time t depends on all previous state descriptions $\mathbf{x}_{t-1}$, as well as on observations $\mathbf{o}_{t-1}$ perceived so far, and actions $\mathbf{a}_{t-1}$ executed.

But, estimating the joint probability distribution over all considered variables of the current world state representation is much too complex in robotics. The following assumptions significantly reduce its complexity:

1. The *Markov assumption*, also known as static-world assumption, reduces the dependence of the current state description to the previous one, $Bel(x_{t-1})$, and the action taken there, a_{t-1}, such that

$$Bel(x_t|\mathbf{x}_{t-1}, a_1 \dots a_{t-1}, o_1 \dots o_{t-1}) = Bel(x_t|x_{t-1}, a_{t-1})$$

that the past is independent to the future given complete knowledge about the current state [Thrun, 2000, p. 3]. This is a very strong assumption that is violated as soon as observable objects besides the agent are present in the environment (cf. Fox et al. [1999, p. 395]). Note that if the state description depends on some observable variable, adding it to the state description already can alleviate the problem of violating the independence assumption, but infers increased complexity. Although it is a strong restriction, real-world applications exhibited some degree of tolerance to it.

2. Different sensors are independent to each other and are in use in Formula 2.1, too.

3. Actions only depend on previous percepts:

$$P\left(a_{t-1} \mid a_1 \ldots a_{t-2}, o_1 \ldots o_{t-1}\right) = P\left(a_{t-1} \mid o_1 \ldots o_{t-1}\right) \qquad (2.4)$$

Then, the general prediction step of Bayesian state estimation given in Formula 2.2 computes the new state vector $Bel(x_t)$ after action a_{t-1} has been performed by:

$$Bel'(x_t) = \int P(x_t | x_{t-1}, a_{t-1}) Bel(x_{t-1}) \mathrm{d}x_{t-1}$$

The second stage incorporates the new percept, o_t, yielding the updated belief $Bel(x_t)$:

$$Bel(x_t) \propto P(o_t | x_t) Bel'(x_t)$$

Taking both stages together, the Bayesian state update is defined by

$$Bel\left(x_t\right) \propto P\left(o_t \mid x_t\right) \int P\left(x_t \mid x_{t-1}, a_{t-1}\right) Bel\left(x_{t-1}\right) \mathrm{d}x_{t-1} \qquad (2.5)$$

This equation shows how two very different inputs, namely the observation o_t and the action a_{t-1} are incorporated into the common state representation $Bel(x_t)$ using Bayesian probability theory.

Inferences of the second step about how agents can act according to the new belief can be obtained using utility and decision theory, that are only mentioned here (cf. Russell and Norvig, 1995).

Belief Networks

The integration of sensory information that refers to the same state variable from different sources can be concisely represented by a Bayesian belief network. The advantage of a belief network is that independence relations among nodes in the graph (state variables) can be easily recognised and exploited in the computation. If a state variable is measured by different sensors, the percepts are independent to each other given the actual value [Russell and Norvig, 1995, p. 512]. In belief networks, the sensor model implements the connection between a sensor and the corresponding state variable.

In order to account for time-dependent state variables, a dynamic belief network considers a standard belief network at two time instances, $t-1$ and t, in the sequential Markov chain.

Since the computation of the new state estimate in both, standard and dynamic belief networks, is based on Bayesian state estimation given above, the reader is referred to Formulae 2.2 and 2.3.

Since probability theory focuses on uncertainty by means of prior and posterior variables, it does not allow for expressing other aspects of uncertainty like for instance ambiguity, ignorance or inconsistency. In order take such aspects into account, an engineer carefully estimates an average over these and represents them altogether by a single value of uncertainty. But the obtained approximation often is insufficient due to the limited expressiveness of a single numerical value of Bayesian probability. The following alternative methodologies try to improve the expressiveness concerning uncertainty.

2.4.2 Dempster-Shafer Theory of Evidence

The main motivation for devising the Dempster-Shafer Theory of Evidence (DST) was to incorporate a means to distinguish between uncertainty and ignorance [Dempster, 1968, Shafer, 1976]. While Bayesian theory concerns the probability of a proposition, in DST the belief function $Bel(x)$ represents the amount of evidence supporting the proposition x.

Concerning expressiveness of uncertainty, there are three important differences between Bayesian theory and DST [Krause and Clark, 1993, p. 68]:

1. In DST, the belief can be assigned to a set of propositions, in contrast to Bayes' theory where each event needs to be assigned a unique, subjective probability.

2. Bayesian theory enforces a strong commitment to the belief in the negation of a proposition, that is $m(x) + m(\overline{x}) = 1$. DST being driven by evidences, both, $m(x)$ and $m(\overline{x})$ depend on the degree of support received from evidences. This accounts for the DST's ability to express a degree of ignorance and disagreement. Hence, the ignorance explicitly represented in DST by $Bel(x)$ has to be implicitly determined in Bayesian theory from the changes in the probability $p(x)$.

3. Dempster's rule of combination represents a means to integrate evidences from various sources, especially regarding inferences from sets or complements thereof.

In the following, those properties of DST that are interesting regarding sensor fusion are highlighted.

Shafer [1976] introduces the notion of a basic probability assignment (BPA), $m(X) : 2^\Omega \to [0, 1]$, as a measure of support for a given set of propositions, X, ranging over the frame of discernment Ω by defining $m(\emptyset) = 0$ and $\sum_{X \subseteq \Omega} m(X) = 1$, where $\emptyset$ denotes the empty set. While $m(X)$ describes the evidential support of individual sets of propositions, the belief function $Bel(X)$ summarises all available evidences for a collection of subsets from the frame of discernment, $X_1, \ldots, X_n \subseteq \Omega$:

$$\begin{aligned} Bel(X) &= \sum_{Y \subseteq X} m(Y) \\ Pl(X) &= 1 - Bel(\overline{X}) \end{aligned} \tag{2.6}$$

Equation 2.6 defines the *plausibility* for a given set X, which represents the extent to which the available evidence fails to refute X [Krause and Clark, 1993, p. 79]. Since $Bel(X) \leq Pl(X)$, the status of the set X is represented by the interval between the available support and the degree of not refuting X: $[Bel(X), Pl(X)]$. This interval provides the means of representing absolute certainty, $[1, 1]$, as well as total ignorance, $[0, 1]$. Note that Bayesian probability theory lacks this expressiveness due to the constraint on assigning a single numerical probability value only.

Dempster's rule of combination defines the integration of two independent evidences, $m_1(X)$ and $m_2(Y)$, into the common representation $Z = X \cap Y$ in [Dempster, 1968] by:

$$m_1 \oplus m_2(Z) = \frac{1}{c} \sum_{X \cap Y = Z} m_1(X) m_2(Y) \quad (2.7)$$

$$c = 1 - \sum_{X \cap Y = \emptyset} m_1(X) m_2(Y) \quad (2.8)$$

While Equation 2.7 defines the support of both evidences simultaneously, Equation 2.8 is required for normalisation. The rule clearly exhibits the fact that the computation of evidential support is based on sets, which provides additional flexibility in the specification and expressiveness.

As another example for differences in expressiveness between Bayesian and DST, in DST it is possible to integrate evidences of total ignorance with no effect on the resulting belief.

Although Dempster-Shafer theory of evidences provides an improved expressiveness of semantics in comparison to Bayesian probability theory, its increased complexity infers some problems. For instance, the normalisation in Dempster's rule of combination causes contentious discussions about its appropriateness (cf. e.g. Clarke [1988]). Another problem occurs when the normalisation denominator becomes very small, which may yield irrational values for the belief [Krause and Clark, 1993, p. 85].

A rather different approach than those mentioned above is the Fuzzy theory.

2.4.3 Fuzzy Theory

In contrast to both, Bayesian probability theory and Dempster-Shafer theory which represent the ontological degree of *belief in the truth* of a proposition X (or a set thereof), Fuzzy logic deals with the *degree of truth* in X. "Although fuzzy logic is a healthy, ongoing enterprise, it is increasingly perceived as a way of handling continuous-valued variables rather than uncertainty." [Russell and Norvig, 1995, p. 467]

The fusion of uncertain information using Fuzzy theory refers to a classification approach and comprises the following steps.

1. First, the value of the percept X matching the classifier C_i for all $i = 1 \ldots n$ classifiers is computed in the matching step by

   $$a_{ij} = \text{match}(X_k, C_{i,j})$$

   where X and C_i are vectors of attributes. A prominent matching operator, match(), is $\max - \min$.

2. Next, the matching values for individual attributes, $a_{i,j}$, are combined by

$$a_i = \text{aggreg}(a_{i,1}, \ldots, a_{i,m})$$

using an aggregation operator, e.g., min.

3. Finally, choosing the best match among a_i determines the object recognised.

Herein, the choices of operators applied depends on the task to solve and have to be verified in experiments.

2.5 Applications of Sensor Fusion

Subsequently, a few applications of the methods given in Section 2.4 and other methods to sensor fusion are described.

The domain of photogrammetry and imaging exhibits a high demand for sensor fusion. One example is the fusion of aerial or satellite images into a global, common map. Due to the vast amount of data, an automatic approach is desired (cf. Gens et al., 1998).

Another application of steady high demand for sensor fusion, is weather forecasting, where a huge amount of data perceived from various sensors of different characteristics needs to be fused into a shared representation (cf. Evans et al., 1999).

A rather simplistic approach to fuse spatial information is called *certainty grid* [Moravec, 1988]. Similarly to an occupancy grid, which has been developed later, the perceptive range of a distance sensor is discretised into regular grid cells. In [Borenstein and Koren, 1991] the certainty grid approach is applied to sonar sensors. The basic version updates the values of those grid cells laying on the acoustic axis at a distance shorter than the measurement by $CV'_{i,j} = \|CV_{i,j} + I^-\|$, i.e., the current value is decremented by a fixed value ($I^- = -1$) and normalised to stay within given bounds $[0, 15]$. Furthermore, the grid cell laying on the acoustic axis at the measured distance is updated according to $CV'_{i,j} = \|CV_{i,j} + I^+\|$ with $I^+ = +3$. Repeating the grid update process at a high frequency, it represents a dynamic probability distribution of occupancy.

The agent's pose being the most important state description in robotics, the Bayesian state estimation is the dominant method employed in robot localisation (cf. e.g. Thrun et al. [1998], Burgard et al. [1998a], Konolige and Chou [1999]). This includes Markov as well as Monte-Carlo approaches, since both are based on the same mathematical background presented in Section 2.4.1. Most of the early implementations of robotic localisation employ the Kalman filter for state estimation.

The middle-size league of the RoboCup domain provides a very different example of sensor fusion. Here, each individual robot sends its percepts to a central computer, which integrates all incoming sensory data into a global representation. Then, the representation is sent back to the agents in the playing field. Perception contains high uncertainty but also a lot of redundancy. The implementation of sensor fusion has a considerable effect on the outcome of the game.

2.5.1 Applications of Bayesian Theory

In Cameron and Wu [1991], a system for identifying and localising electrical components is described that takes uncertain information in the world representation into account for planning active sensing. They experienced that although conflicting data was perceived, the "Application of the Bayesian theorem in such cases has proven successful in overcoming this challenge." [Cameron and Wu, 1992]

Gorodnichy [1999] states that both, the DST and the Bayesian approach produces undesired results, which was his motivation to employ an Adaptive Logic Network instead. This observation is based on empirical comparison between Bayes, Dempster-Shafer and Fuzzy theory. Using Bayes' rule in the Bayesian framework, he defines the combination of two evidences, r_1 and r_2 (range measurements), of the same world point, r, to be **o**ccupied by:

$$P(o|r_1, r_2) = \frac{P(r_2|o, r_1)P(o|r_1)}{P(r_2|o, r_1)P(o|r_1) + P(r_2|e, r_1)P(e|r_1)} \tag{2.9}$$

A very similar approach can be found in [Ribo and Pinz, 2001], where an occupancy grid representation is updated from sonar sensor measurements. Their Bayesian state update using the same notation as above simply is:

$$p(o|r) = \frac{p(r|o)p(o)}{p(r|o)p(o) + p(r|e)p(e)}$$

The initialisation of all grid cells with $p(o) = p(e) = 0.5$, i.e. "unknown", exhibits the weak expressiveness of Bayesian probability theory with respect to ignorance.

If the strength of the constraining independence assumptions often cited above is a major concern, Gutmann et al. [1999, p. 406] provide an example how to still rely on the Markovian property of the agent's environment. Being a rather ad hoc solution, they alleviate the influence of dynamic objects on perception by employing an Entropy filter to discard dynamic objects.

A feature-based approach to sensor fusion is followed in [Castellanos et al., 2001]. Here, two different sensor devices (CCD camera and laser range finder) observe similar portions of the environment. In this work, common features among both sensor information (corners) are extracted (see p. 25 for a discussion on features of this work). According to the process of sensor fusion explained in Section 2.3, data alignment is performed first reducing both perceptive volumes to the shared portion. Next, data association is performed by selecting correspondences based on the Nearest Neighbour approach using the Mahalanobis distance. It is worth noting that "this step benefits from the availability of an accurate sensor-sensor calibration." [Castellanos et al., 2001, p. 910] Finally, measurements are integrated into the internal representation using a standard Extended Kalman Filter. The involved uncertainties are integrated and propagated accordingly.

2.5.2 Applications of Dempster-Shafer Theory

According to [Gorodnichy, 1999], the Markov assumption regarding independence among different evidences is violated in Equation 2.9. Employing Dempster-Shafer theory alternatively, the combination of evidences for the considered point being **o**ccupied $m(o)$ or **e**mpty $m(e)$ is obtained from:

$$m(o) = \frac{m_1(o)m_2(o,e) + m_2(o)m_1(o,e) + m_1(o)m_2(o)}{1 - m_1(o)m_2(e) - m_1(e)m_2(o)}$$

The denominator refers to Equation 2.6, which computes the uncertainty interval, $1 - m(e) - m(o)$ in the case of a single evidence. By that, a degree of ignorance versus contradiction can be expressed. Again, according to Gorodnichy [1999] the independence assumption among evidences does not hold either in the DST framework.

Besides a Bayesian approach mentioned above, [Ribo and Pinz, 2001] also implemented a sonar sensor model for the evidential DST. Disregarding the details of the specific sensor model, the update process according to Dempster-Shafer conforms to Equations 2.7 and 2.8, where the current belief in the state of a given occupancy grid cell represents the one evidence and the percept the other one. All grid cells are initialised with $m(o) = 0$ and $m(e) = 0$ exhibiting total ignorance.

2.5.3 Applications of Fuzzy Theory

Enderle et al. [2000] describe a vision-based robot localisation method for the RoboCup domain. They perform matching of extracted features from camera images with a world model using a probability distributions for each expected feature location. The matching process can be seen from the Fuzzy theory perspective, because basic, triangular matching distributions for individual features are simply combined in the manner of object classification. This work resembles the information fusion method in Fuzzy theory (Fuzzy integration) strongly, which is described in Section 2.4.3 above.

In the empirical comparison of Ribo and Pinz [2001], they also explain the adaptation of their sonar sensor model to sensor fusion employing Fuzzy theory besides a Bayesian and DST approach shown above. Similarly to the Dempster-Shafer approach, different sensor models are necessary for computing the evidence supporting emptiness and occupation. Then, the update equations for the state of given grid cell C to be **o**ccupied or **e**mpty upon percept S are:

$$\begin{aligned}
\mu_o(C,M) &= \mu_o(C,M) + \mu_o(C,S) - \mu_o(C,M)\mu_o(C,S) \\
\mu_e(C,M) &= \mu_e(C,M) + \mu_e(C,S) - \mu_e(C,M)\mu_e(C,S)
\end{aligned}$$

where μ denotes the membership function and M the current belief of the internal state representation. All grid cells are initialised to exhibit total ignorance, $\mu_o(C,M) = 0$, $\mu_e(C,M) = 0$. See Ribo and Pinz [2001] for more details on an empirical comparison between the three prominent methodologies described here. In summary, no “best” approach can be determined, but each exhibits its individual characteristics.

Chapter 3

Previous Work on Sensor Models

This chapter briefly reviews some research results on building camera sensor models in particular.

In the early 1980s it was impossible for an autonomous mobile robot to navigate in a cluttered indoor environment. Later, the system FINALE exhibited an example of impressive achievements in vision-based navigation [Kosaka and Kak, 1992]. This system computes regions of interest of a given image, where predicted features are likely to be found. The size of these regions depends on the uncertainties considered, e.g., uncertainty in robot motion. Besides a geometric representation of the agent's environment, the model contains topological information, too. This enabled the FINALE system to navigate at an average speed of 17 m/min running on an off-the-shelf computer of that time without any special hardware.

3.1 Vision-Based Navigation

The control of mobile agents using vision devices in real-world environments has been a prominent and demanding task for the last 20 years. The research on vision-based navigation is separated into two fields: indoor and outdoor navigation with fairly different aims and problems. Here, I will briefly describe the main research topics involved in indoor vision-based navigation.

It was recognised early that a vision-based system for mobile robot navigation requires a priori information about its environment in order to know what its sensors are expected to perceive. Concerning the degree of given knowledge, three branches exist within the field of vision-based indoor navigation. First, *map-based* navigation systems make use of a geometric or topological representation. Second, *map-building-based* approaches employ sensors for creating an internal representation of the perceived environment and use it for navigation at the same time. This branch is also known as Simultaneous Localisation and Mapping (SLAM). The term *localisation* refers to the general method of figuring out where the agent is by means of its percepts and perhaps some additional a priori knowledge, e.g., a map.

Finally, *map-less* approaches have been devised that do not maintain explicit world representations, but rather identify and track landmarks and objects. According to the project at hand, here I focus on the branch of map-based systems, because there are considerable difference among all three methods.

Typically, vision-based localisation comprises the following computation steps (cf. Hanek and Schmitt [2000], Kosaka and Kak [1992]):

1. to obtain the sensory information from the camera;
2. to compute the uncertainty in camera position;
3. for each predicted feature, compute a region of interest in the image, which poses a constraint on feature extraction;
4. to extract certain types of features from the image;
5. to determine correspondences between the perceived and the model features; and
6. to compute or update the new position based on the correspondences found.

Hereof, searching for correspondences turned out to be the most difficult step. Usually, uncertainties of the world representation, the perception process, the robot motion etc. are to be specified and used to determine regions of interest in the image plane, where percepts are predicted to be found and to which feature extraction and correspondence search are constrained.

The FINALE system is a typical approach of vision-based localisation using a geometric world model [Kosaka and Kak, 1992]. Here, the uncertainty in the robot's pose (x, y, θ) is represented by three Gaussian probability distributions. On each discrete motion step, the pose hypothesis is updated according to the issued motion command and the robot's odometry model, which is also known as the prediction step (see Section 3.3). Next, model features are projected to the image plane. Furthermore, uncertainties are represented by elliptical regions centered at each feature's expected image position. By that, the correspondence search is constrained to regions of interest, while their sizes depend on the degrees of involved uncertainty. All possible correspondences are determined among the features within the region constraints. Finally, the mean and the covariance of the hypothesis in the robot's pose are updated employing the Kalman filter according to the degree of matching correspondence [Kalman, 1960].

For that, all required mean values and standard deviations were obtained from experiments, in particular the parameters of the odometry model. It is questionable, whether the Kalman filter limited to Gaussian distributions for all involved probabilities is appropriate. The constraints on the search space for image features determine the region around a predicted feature where the measured feature may be found up to a certain degree of tolerance only. Taking all regions together may already cover a high portion of the image, especially in the case of feature-rich images and high uncertainties. Hence, if the constraints on processing within the image should be of advantage, then a fairly accurate pose hypothesis is required. Please note that the precision and appropriateness of the involved sensor models (odometry and camera sensor) play a key role for successful establishment of correspondences, since they influence accuracy and tolerance of the matching and update stage. Likewise to my proposals concerning the type of features to use and the coordinate system of matching given in Section 4.6, the FINALE system extracts straight line features and performs the matching process in the Hough space exploiting the concise representation of line segments possible there.

Hanek and Schmitt [2000] followed the classical approach to visual localisation and successfully applied it in the domain of RoboCup soccer having a well-defined playing field with distinctive features. The world model comprises curve and line features, but curves are approximated by line segments eventually. The distance along the normal vector between model and image lines in the image plane is used for evaluation of displacements of features, similarly to many other implementations (cf. Section 3.4). Here, Gaussian noise of the perceived features and Gaussian uncertainty in the robot pose with zero mean are assumed. Then, the robot's pose is iteratively optimised. The boundaries of regions around predicted feature locations defined by the propagated uncertainties are used to constrain the search for image features and correspondences for the sake of performance.

A simple approach to vision-based self-localisation is described in [Collins and Beveridge, 1993]. Features are assumed to refer to the same coplanar structure as typically found in human-made environments. First, the expected frontal view is generated using the world model. Next, the relative location of the 3D plane under consideration is computed by determination of some vanishing points of the scene using the Hough transform and rectification with the vertical image axis. Then, the best matching affine transformation is computed by local search, which directly determines the robot's pose relative to the 3D structure at hand. This approach is rather different to the typical procedure described on page 20, but confirms that straight line segments are the most prominent kind of features suitable for indoor feature-based navigation, which is employed in the project at hand, too.

Some vision-based approaches are based on the weak perspective camera model (cf. Huttenlocher and Ullman, 1990). Since this model employs orthographic projection and scaling such that objects in the world are assumed to have the same depth, it is a rough approximation and hence only applicable in small ranges concerning the image size and is not considered here. In contrast to that, the project at hand employes a non-linear projective camera model and considers second order radial distortion in the vision pre-processing stage.

3.2 Localisation Using Time-of-Flight Sensors

Common time-of-flight sensors employed in robotics are infra-red sensors, sonar sensors and the laser range finder. In accordance to their considerable differences in operation and percepts in comparison to vision sensors, the corresponding techniques differ as well. Nevertheless, they form a similar sensor model and are therefore subject of this section.

The *occupancy grid* is a prominent representation type of the world model, which was originated by [Moravec and Elfes, 1985] under the certainty grids. Here, a (usually regular) grid discretises the world representation into cells, each bearing a value for the degree of spatial occupancy of the corresponding volume in the world. The occupancy grid was extended with the capability to represent uncertainties of measurement later. Hereof, the basic idea is to maintain two separate cells for each discrete world cell: one reflecting spatial occupancy and one reflecting spatial vacancy, similarly to a certainty grid. Examples for applications using this grid representation are the Vector Field Histogram approach [Borenstein and Koren, 1989] and Fuzzy operators [Oriolo et al., 1995].

Approaches to *global localisation* mostly differ from incremental localisation (*position tracking*) in their ability to represent an arbitrary probability density function (PDF) that can contain multiple pose hypotheses simultaneously, as opposed to an unimodal PDF like the Gaussian distribution. Approaches to global localisation have to figure out the agent's pose in the whole environment from scratch solely based on percepts from sensors and a map-like representation of the environment (cf. Fox et al., 1999). In addition, approaches to position tracking require an approximate pose hypothesis to be given, which is subsequently propagated and refined (cf. Burgard et al., 1998b).

Using simple geometric entities in the world representation instead of unique landmarks, global localisation has to cope with highly ambiguous hypothesis matches. Multiple hypotheses have to be appropriately propagated to future hypothesis checks until some of them can be confirmed, while others can be discarded if insufficient perceptive support is observed. Besides basic solutions like representing uncertainties in intervals (cf. Atiya and Hager, 1993) and triangulation (cf. Sugihara, 1988), other methods are more suitable for multi-modal hypothesis tracking like Partially Observable Markov Decision Processes (cf. Simmons and Koenig, 1995), Markov localisation using occupancy grids (cf. Fox et al., 1999), Monte Carlo localisation using sample sets (cf. Fearnhead, 1998), or Kalman filtering based on a mixture of Gaussian distributions (cf. Cox and Leonard, 1994). Roughly speaking, methods appropriate for global localisation are applicable for position tracking, too.

3.3 Sensor Model

Recalling the requirement of an internal model of the world by vision-based navigation, Crowley [1995] calls maintaining this representation "dynamic world modeling". This representation is frequently updated upon arrival of sensory information of the agent, be it percepts from distance sensors, visual devices or odometry sensors. In order to integrate different percepts into a commensurable representation, a specific sensor model is needed for each individual kind of sensor device. A *sensor model* defines the entire sensing process, such that observations can be converted to conform to the world model representation, and expectations can be predicted vice versa given the world model using the inverse of the sensor model.

The processing of sensor models differs in the level of data abstraction of the resulting primitives, i.e., the extent of semantic interpretation, which are for instance raw signals and features. The amount of data pre-processing and interpretation necessary in order to gain commensurable primitives comprising common properties for primitives of the world model and of the percepts, strongly depends on the type of sensor and the world model's abstraction level.

Concerning the aim of the project under consideration, the region common to both sensors, laser range finder and CCD camera, is the three-dimensional space defined by the intersection between the regions of view of each sensor and roughly resembles the scanning plane of the laser range finder. Similarly to the work of Castellanos et al. [2001], commensurable features of both sensors are obtained if corners are extracted from the segmented laser measurements, and vertical line features are recognised in camera images.

According to the semantic abstraction level, there exist two major groups: dense sensor techniques that directly use raw data (cf. Moravec [1988], Burgard et al. [1996], Dellaert et al. [1999], Konolige and Chou [1999]), and feature- or landmark-based techniques that extract features from perceptive measurements (cf. Kortenkamp and Weymouth [1994], Simmons and Koenig [1995]).

For example, the general Bayesian state update scheme given in Section 2.4.1 is used in the Kalman filter and in Markov localisation and is repeated here:

$$Bel\left(x_t\right) \propto P\left(o_t \mid x_t\right) \int P\left(x_t \mid x_{t-1}, a_{t-1}\right) Bel\left(x_{t-1}\right) dx_{t-1} \tag{3.1}$$

Here, the new belief in hypothesis x_t at time t results from all possible previous beliefs, $Bel(x_{t-1})$, weighed by their transition probability to reach x_t from x_{t-1} by an action a_{t-1}. All possible transitions are weighed by the sensor model $P\left(o_t \mid x_t\right)$ of perceiving measurement o_t at position x_t.

This shows that two (joint) PDFs have to be computed for each incremental update of the current belief, $Bel\left(x_t\right)$, given the previous hypothesis $Bel\left(x_{t-1}\right)$:

$P\left(o_t \mid x_t\right)$ The perception model defines the probability of the sensor measurement o_t at location x_t. Exploiting the Markov assumption, the perception model (sensor model) is simplified to

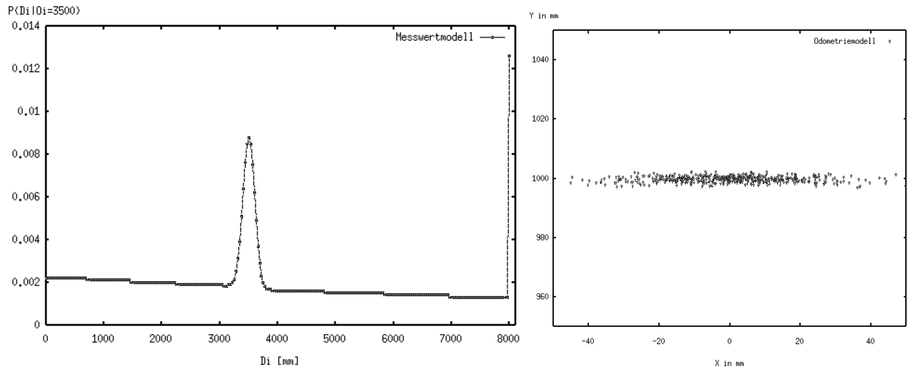
$$P(o_t \mid x_1 \dots x_t, o_1 \dots o_{t-1}, a_1 \dots a_{t-1}) = P(o_t \mid x_t)$$

$P\left(x_t \mid x_{t-1}, a_{t-1}\right)$ The odometry model determines the probability of reaching location x_t from x_{t-1} when the motion command a_{t-1} was issued before.

3.3.1 Distance Sensor Model

The most common perceptive sensor devices used in robotics are sonar sensors, laser range finders (LRF) and CCD cameras. While the sensor model of a single sonar device has to model the measurement process within the perceptive cone at decreasing certainty at longer ranges and at higher angular deviations from the main axis, the sensor model of laser range finders can be approximated by a mixture of three PDFs. Figure 3.1(a) depicts the measurement model representing the probability $p(d_i|o_i = 3500\,\mathrm{cm})$ of measuring distance d_i given an expected range measurement of distance $o_i = 3500$ cm. The correct percept is modeled by a Gaussian distribution centered at the perfect measurement ($o_i = 3500$ cm). In addition, a uniform probability represents the presence of noise and dynamic objects, both in front of and behind the expectation. Furthermore, a high probability summarises the possibility of measuring a value greater than the maximum range ($o_i \geq 8000$).

Usually, the combination of these PDFs is pre-computed and sampled yielding a lookup table. This distance sensor model proved to be appropriate in own experiments adopted from [Fox et al., 1999], where its mathematical foundation is given.



(a) Distance sensor model (LRF) with expected distance 3500 cm

(b) Motion model: predicted position after one meter of straight forward motion

Figure 3.1: Examples of sensor models

The inverse process of LRF measurements is usually obtained by ray-tracing in the world model. Ribo and Pinz [2001] provide a comprehensive discussion on mathematical sonar sensor models.

3.3.2 Odometry Model

The odometry model (motion model) given in Fox et al. [2001, p. 473] models the uncertainty in robot motion by a convolution of two independent Gaussian random variables with zero mean at the perfect prediction representing uncertainty in translation and rotation. In a former student project I employed an additional Gaussian random variable for representing lateral drift. This is depicted in Figure 3.1(b), where a set of samples was drawn at random according to the PDF of the odometry model. In particular, the Figure shows the probability $p(x_t|x_{t-1}, a_{t-1})$ of ending in pose $x_t = (x, y, \theta)^T = (0, 1000\,\text{cm}, 0)^T$ starting from $x_{t-1} = (0, 0, 0)^T$ when the action $a_{t-1} = (+0, +1000\,\text{cm}, +0)^T$ has been performed, which represents a straight forward motion of distance one meter). Note that Figure 3.1(b) actually illustrates a two-dimensional projection disregarding the robot's orientation θ for the sake of visualisation. Using three zero-centered Gaussians is a prominent description of the robot's odometry error in the literature (cf. Gutmann et al. [1998, p. 4]).

While this models the general, uncertain process of robot motion applicable to many types of mobile robots, the accuracy of the whole navigation task can be improved using specific parameters for each individual robot that have to be obtained from experiments. Unfortunately, those parameters are only valid within the specific environment due to very different floor properties, which can be circumvented by a rough motion model with high tolerances.

3.3.3 Camera Sensor Model

Compared to the sensor models mentioned above, it is rather difficult to define a model of a visual sensor. The classical approach usually followed in monocular vision-based self-localisation (see page 20) defines a camera sensor model implementing a particular method.

In the work of [Castellanos et al., 2001] a visual sensor is employed for providing a current feature map on demand. Here, the result comprises the feature map and a corresponding covariance matrix that reflects the uncertainty in geometric location of each entity. This information can be seen as being the interface of their camera sensor model between the sensor specific data processing and the commensurable feature frame. Since this approach aims at sensor fusion of perceptive information from a camera and a laser range finder, the common representation is defined by the horizontal plane of the LRF that coincides with the camera viewing volume. Corner features are their choice of common features among the camera and the laser feature map bearing a common set of properties. Hence, vertical line segments are extracted from video data forming rays to corners at the intersection with the laser scanning plane. Corner features from laser scans are obtained by a segmentation algorithm. Hereby, perceptive grouping of features yields a higher feature abstraction level, which in turn results in a more robust environment description. This shows a kind of a camera sensor model that works in polygonal indoor environments and employs a specific type of features in accordance to the second sensor type (LRF) in use.

An example for a very environment-specific camera sensor model for a mobile robot can be found in [Enderle et al., 2000]. Here, a colour camera was exclusively employed for self-localising their mobile robot on the well-defined soccer field in the RoboCup domain. For the sake of easy recognition, the goals bear different colours, the floor has a unique colour, the walls have a specified colour, and the corners are marked with vertical bars of unique colour. Moreover, all sizes and locations of world objects and colour segments are well known. Enderle et al. [2000] implemented a probabilistic, feature-based camera sensor model, that extracts these particular features from images. Then, the position of each measured feature is matched according to a simple probability function of triangular shape with the corresponding predictions given the world model and the current pose hypothesis (see also Section 2.4.3). By that, a probability function is formed conforming to Equation 2.5. The sensor model is employed to weigh each sample in the Monte Carlo localisation update steps.

3.4 Similarity Measure

This section points to some examples of previous research from different fields of technology like object recognition and tracking, mobile robot navigation and camera observation, which particularly relate to a vision-based sensor model.

The comparison of raw pixel data is only applicable when the degree of changes between both scenes are very small, which depends on many factors like illumination, presence and reflectance properties of structures in the surroundings and objects perceived, the location of the camera, etc.

An example for a similarity measure based on a signal-level feature is the colour histogram. The similarity between image I and J, where i be the feature representing the number of pixels i_k of colour k of image I (j similarly) can be defined as simple as

$$s(I,J) \sim 1/\sum_k |i_k - j_k|$$

which yields a commensurable scale if normalised to the number of colours and pixels in use. If instead two regions are to be compared, the correlation measure given by

$$c(\delta x, \delta y) = \sum\sum f(x,y)g(\delta x + x, \delta y + y)$$

sums up the differences of the intensity values $f(x,y)$ and $g(x,y)$ over both image regions considered.

When tolerances to changing illumination conditions in dynamic environments are desired, a more invariant description of the image ought to be obtained by extraction of features. Subsequently, research on or using similarity measures is reviewed and compared with those of our project.

Lowe has done some considerable research on vision-based pose estimation of 3D objects (cf. Lowe, 1987). He extracts line segments from a video stream and optimises the pose parameters with six degrees of freedom (DOF) using least squares optimisation. For that, a similarity measure is employed that is based on the distance $d_{2D,1}$ defined on page 28. Likewise to Lowe's work, the sensor model devised in the project at hand employs line segments as the entities of geometric scene recognition and computes the probability of the percept, which various applications like localisation or sensor fusion can make use of. Being an optimisation process, Lowe's pose estimation approach relies on a pose hypothesis to be given a priori. In opposite to my approach, distance measures evaluate feature correspondences that have to be searched for (cf. Lowe [1987], Lanser [1997]). This implies a great computational demand, which is inavitable if topological structures are to be extracted and compared according to semantic criteria, e.g., parallelism, proximity or collinearity. My implementation employs a distance measure without requiring correspondence search and image feature interpretation. Similarly to the project at hand, where the distance measure operates in the Hough space, the correspondence search in Lowe's approaches is controlled by

1. the distance between image and projected model lines along the normal vector of an image line coinciding at the line center (denoted normal distance for short), and

2. the deviation in orientation of the image line with respect to the projected model line.

Lanser [1997] recognises a major problem of Lowe's approach in the initial pose estimate for optimisation, because the differences in distance and orientation differ considerably when changing the viewpoint. Nevertheless, Lanser grounds his correspondence search on similar measures, but enhances the correspondence verification using several relations on the topological level among groups of features. In contrast, the sensor model proposed in the report at hand actually exploits the circumstance that small changes in the viewpoint can be easily recognised in the projected image (see Section 4.6).

Du et al. [1993] often obtained bad results applying the early approaches of Lowe. They report that this is mainly caused by truncation of lines in image plane (due to bad contrast and occlusion), noise (causing correspondence problems) and clutter. Their solution for the similarity measure between an image line and a projected model line was a weighed sum of the difference of orientations, the normal distance and the deviation in line length, where the first was weighed most.

In the project at hand, truncation is a major concern. The best result was achieved using a simple similarity (distance) measure solely based on a match count, while the version taking the lengths into account scored well, too. Besides the perturbation effect of split lines, truncation motivated the choice of the Hough space as the coordinate frame of comparison, because in this representation line segments still remain at the same position if truncation or splitting occurs. Furthermore, noise is first dealt with in the image pre-processing stage by convolution with a smoothing mask, and second in the feature extraction stage. Here for instance, the recursive line regression finder takes a threshold parameter for the minimum length of line segments to extract.

Both, Wunsch [1997] and Lanser [1997] employ the following distance as the similarity measure:

d_{3D} — denotes the distance between the endpoints of model lines in the camera coordinate system and the "view plane" that reaches through the image line and the camera projection center.

This distance is analogous to our similarity component in the image plane based on the difference of normal distances in the Hough space using the projection of model lines. Besides line segments, they suggest using points and ellipses as features, but only lines are actually implemented.

Daucher et al. [1993] describe a vision-based system for tracking objects that searches for correspondences between the features extracted from video measurements and those obtained from the model. The correspondes are evaluated using the distance measure d_{3D} adopted from Lanser. They compute the object pose estimate analytically instead of optimising an initial pose guess in opposite to the other approaches mentioned here.

Lanser [1997, p. 106] considers many kinds of uncertainties:

- Uncertainty of the pose hypothesis, which is the dominant uncertainty;
- Spatial inaccuracy of the model, mainly due to measurement mistakes and approximation of surfaces;
- Inaccuracy of the relative position of the vision system to the mobile agent;
- Inaccuracy of the intrinsic camera calibration parameters;
- Uncertainty of features obtained from video measurements.

These uncertainties are partly propagated to considerations in the image plane and mainly influence the extent of the search for correspondences. First, it is questionable whether all these uncertainties, required to be known in advance, can be appropriately specified. Second, the assumption of Gaussian distributions for all uncertainties does not necessarily hold. For instance, often ambiguities of pose estimates are present due to similar patterns in the world model, which can be only represented appropriately by multivariate distributions.

Hauck and Stöffler [1996] extract line features from camera images in order to determine the states of joints of an articulated object. On top of that, corner features are formed. By addition of a "detectibility" attribute, they achieve a kind of dynamic model description. The state is determined by searching for the best correspondence between the feature map of the video measurement and the model projection. For that, the evaluation is based on the differences of the lengths and angles, and the normal distance between the image line and projected model line. Analogous to our observations, the extraction of line segments from images yields a rather invariable feature with respect to different illumination conditions and lighting effects.

Kumar and Hanson [1994] pursued investigations on video-based localisation using line features. Similarly to the approaches mentioned above, they determine correspondences between the video measurement and the generated model view. Each correspondence is evaluated by the distance measure d_{3D}. Because this measure strongly depends on the depth of the model line, Kumar and Hanson [1994] and Lanser [1997] employed the projected versions of this distance measure later:

$d_{2D,1}$ is based on the 2D distance between the endpoints of the projected model line in the image plane and the image line of infinite length;

$d_{2D,2}$ is based on the 2D distance between the endpoints of the measured (image) line segment and the projected model line of infinite length.

Lanser [1997] showed in simulations that both measures yield similar results on real-world data, but he preferred $d_{2D,1}$ due to its better efficiency of computation.

Beveridge [1993] uses a similar three-dimensional distance measure as d_{3D} for comparing line feature maps extracted from the image and the environment model. Here, lines bear an additional attribute that represents the sign of the originating edge gradient. The search for correspondences is guided by both, a rewarding similarity measure based on matched features and aother measure punishing features that do not match. In his work, Beveridge concentrates on efficient local search methods for correspondences.

The similarity measures of the project at hand described in Section 4.6 do not punish features that do not match on purpose, because the sensor model should be able to tolerate a certain degree of discrepancy between the model and the measurements. This concerns approximations in the world model as well as dynamic objects (noise) not present in the static world model.

Apparently, most pose estimation approaches employ a computationally expensive correspondence search lead by similar distance measures. In contrast to the project at hand, performing object pose estimation assumes to deal with a single object model that is completely visible by the camera. The mobile robot being subject of this report operates in an environment of large scale with respect to its own size. No correspondences are established — instead a degree of similarity between the measured scene and the belief in the current pose given an environment model is computed at the geometric feature level. Anyway, the devised sensor model provides the opportunity to build applications like pose estimate optimisation on top of it, as pointed out in Section 6.1 below. Furthermore, the tolerances implicitly represented by the sensor model alleviate from dealing with occlusions of model features in our environment model.

Chapter 4

A Feature-Based Camera Sensor Model

As has been pointed out in Section 3.3, a sensor model is required in sensor fusion for each sensor employed in the fusion process. This chapter describes the development of a monocular camera sensor model that is appropriate for sensor fusion in our project. The sensor model extracts features from both, the camera image and the world model in order to yield invariance to illumination conditions. It is a generic model for monocular CCD cameras given a specification and office environments given the corresponding geometric world model.

4.1 Problem Specification

The target architecture is a Pioneer 2 DX mobile robot made by ActivMedia[1], which is equipped with various sensors: a front sonar ring comprising 8 sensors, a magnetic compass, a Sick laser range finder (LRF) and a charge-coupled device (CCD) camera. In this project, we focussed on investigating the suitability of employing the camera for geometric navigation.

The objective of the sensor model is to gain a degree of confirmation or rejection on the robot's belief about the current, three-dimensional pose solely based on visual information. As is illustrated in Figure 4.1, the sensor model takes a camera image m and the robot's current position hypothesis l as inputs, and returns the probability for this percept given the location and a model of the environment w. The world model is required for the comparison between measurements (images) and expectations. In this project, a geometric, polygonal model has been generated, which should be appropriate for indoor office environments. A three-dimensional view from the perspective of the robot within the world model is illustrated in Figure 4.6, for example.

The hypothesis of the robot's (camera) pose is used as an estimate, upon which the synthetic camera view within the world model is generated representing the visual prediction (expectation). Subsequently, the correlation between the expectation and the actual measurement is investigated.

[1] ActivMedia Robotics LLC., *ActivMedia Robotics*, `http://www.activmedia.com`, [Online], 08/07/2002.

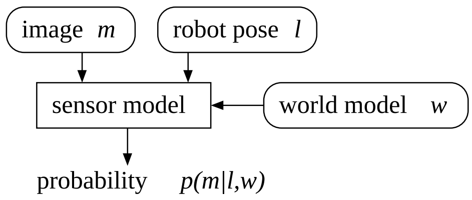


Figure 4.1: Structure of the sensor model

4.2 Image Features

A metric Cartesian system seems to be most appropriate for the input data of the sensor model concerning autonomous navigation. Before the preparation of a three-dimensional model of the office environment that our robot lives in, an important question comes up: Which kind of features are to be used in the recognition and matching process of the sensor model. Without a doubt, the comparison of raw image data is not suitable here, since the changes in illumination, camera position and even scene structure (dynamic objects) are too big. The aim of the project includes a rather robust sensor model applicable in common office environments under regular conditions. Therefore, the extraction of features from the camera data is a necessity in order to yield an invariant description in terms of lighting conditions, dynamic objects in the world, and most importantly, pose deviation, for instance.

Hence, some desired properties of the sensor model to be devised regarding its inputs are:

- Robustness: Clearly, the conditions under which the same scene is observed by the camera can vary drastically in terms of pixel data leading to very different images. Ideally, the result of the sensor model should not be affected by this.

- Independence: The assumption of the model-based matching process to always find all expected landmarks in the measurement and vice versa is too strong for realistic environments. On one hand, occlusion effects, lighting conditions and dynamic objects yield unexpected measurements, both missing and additional items; on the other hand, a world model containing the most distinctive and static landmarks only is usually preferred for the sake of performance. The degree of similarity returned by the sensor model should not depend on the number of discernible items.

- Complexity: The computational costs should be as little as possible. This strongly depends on the kind of features chosen and their extraction methods.

Having these criteria in mind, we applied various prominent feature extraction filters to typical images using the comprehensive image processing package HORUS [Eckstein and Steger, 1997]. Examples, which the appropriateness was tested of, are the Sobel filter, the Kirsch operator, the Laplace operator, the Roberts filter, and the Robinson filter. According to the architecture of our corridor comprising large segments of the same colour, little or no texture and

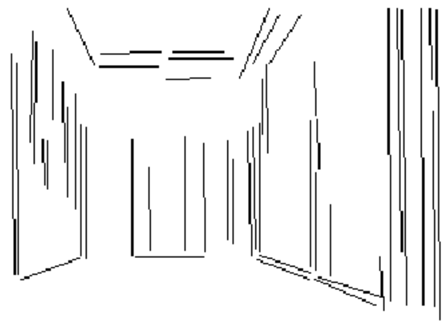


Figure 4.2: Example for the result of line feature extraction.

mostly rectangular shapes, edge filters performed best reflecting the geometry of both, walls and objects.

On top of that, a line segment generation algorithm yields a very compact representation, which is still accurate regarding our objective of navigation. Furthermore, line segments based on image gradient edges allow for a certain tolerance in terms of changing illumination conditions. Please note, that the decision, which kind of features to employ, has been based on feature extraction results obtained from real-world experiments in the target environment.

A typical view of the robot as seen from its perspective employing the CCD camera mounted on top is depicted in Figure 4.5(a). The result of extracting line features from this camera image is illustrated in Figure 4.2, which exhibits the dominance of rectangular shapes regarding objects and walls.

4.3 Camera Model

In order to yield commensurable coordinate systems for both, the camera image and the generated world model, a full geometric model for the camera projection is inevitable. Exhibiting good quality, the employed CCD camera almost conforms to a linear pinhole camera model. Figure 4.3 illustrates the coordinate frames that are involved in the projection process.[2] Here, 3D world coordinates become projected onto the 2D image coordinate frame in a linear manner. The perfect pinhole camera model is based on triangle similarity as given in Equations 4.1 in camera coordinates and is depicted in Figure 4.4.

$$\begin{aligned} x - c_x &= f_x X/Z \\ y - c_y &= f_y Y/Z \\ f_x &= f/p_x \\ f_y &= f/p_y \end{aligned} \tag{4.1}$$

Here, $(c_x, c_y)^{\mathrm{T}}$ defines the principal point, i.e., the location of the image center, while f_x and f_y denotes the so-called focal distance, which is derived

[2] With kind permission from Philip McLauchlan.

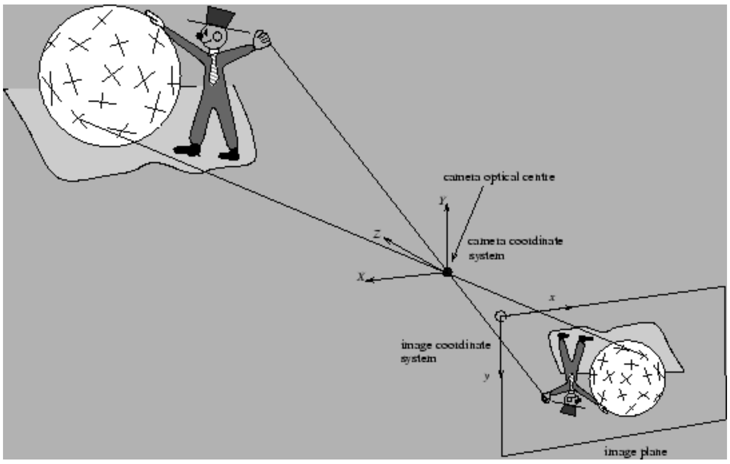


Figure 4.3: Linear pinhole camera model.

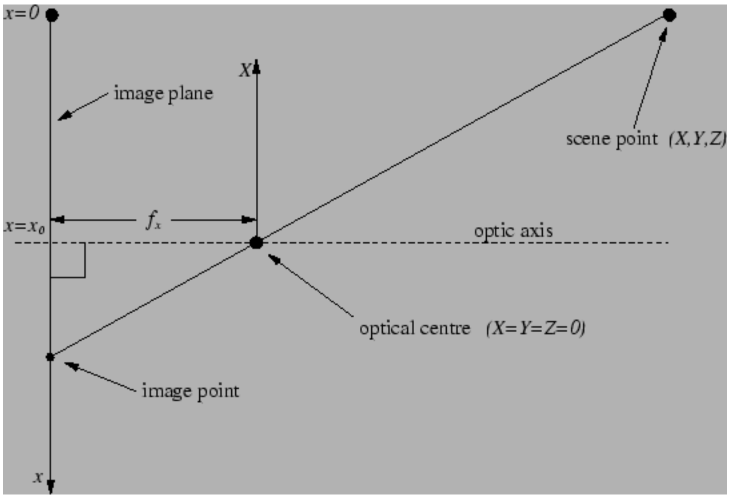


Figure 4.4: Geometry of linear camera model.

from the focal length, f, by normalising with the pixel width, p_x, and pixel height, p_y, respectively.

Although the linear camera model is already a suitable approximation for our Sony camera, one has to take non-linear distortions into account in order to gain better accuracy and generality. This can be achieved by employing the following non-linear camera model instead of the linear model

$$\begin{aligned} x - c_x &= f_x(1+d)X/Z \\ d &= K_1 r^2 + K_2 r^4 \\ r^2 &= (X^2+Y^2)/Z^2 \end{aligned} \tag{4.2}$$

where K_1, and K_2 are the first two parameters of the radial distortion camera model with

$$d = K_1 r^2 + K_2 r^4 + \ldots$$

Again, the equations for y-coordinates look similar. Obviously, the intrinsic parameters f, K_1, K_2, x_0, y_0 are necessary. These are either known from technical specifications, or have to be determined by calibrating the camera. In our case, the camera has been calibrated first according to the Direct Linear Transformation (DLT) method [Abdel-Aziz and Karara, 1971], and second employing the PhotoModeler[3] software. While the DLT approach relies on the linear camera model, the PhotoModeler supports radial distortion with four parameters.

After eliminating the distortion from camera images, they can be directly compared with expectations based on the world model in the same metric coordinate frame.

Because the CCD camera is mounted on our mobile robot, it is connected to an on-board video link, which transmits the video stream to another, stationary video link. From here, a frame-grabber receives the camera data and provides image frames to the user program. While providing great flexibility, this configuration sometimes causes perturbations in the video stream. Two major perturbations were recognised:

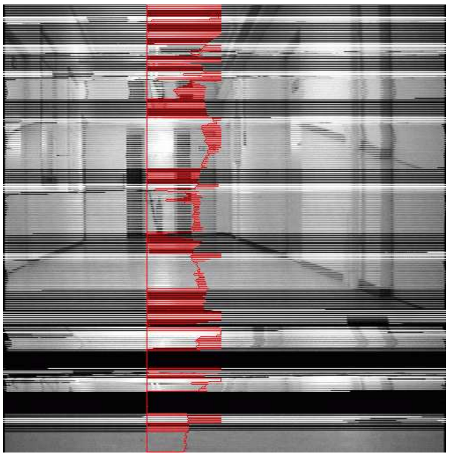
- The quality of the radio transmission depends on the distance between the link devices, and the nature of the environment in between. It typically results in a certain grey or colour value replacing subsequent pixel positions in the sequential chain of image rows, where this effect starts at an arbitrary image coordinate and can last from a few pixels only to 20 and more rows. Figure 4.5(b) shows a severely distorted example, while 4.5(a) depicts regular camera images at the corresponding robot pose. Clearly, these perturbations have to be dealt with, since they will especially affect the edge filter results. Our solution is to consider a vertical amplitude histogram and to compute the squared differences in grey values g for each tuple of pixels of adjacent image rows $\langle(i-1,j),(i,j)\rangle$. If this sum[4] exceeds a specified threshold, that has been determined in experiments, then the image is rejected:

$$\text{reject if}: \; h(j) = \frac{1}{N_y}\sum_{i=2}^{N_y}\left(g_{i-1,j} - g_{i,j}\right)^2 > t \tag{4.3}$$

[3]Eos Systems, PhotoModeler Photogrammetry Software, `http://www.photomodeler.com`, [Online], 09/07/2002.

[4]The square of the L_2 norm can be omitted for performance reasons.

(a) Camera image of our corridor.



(b) Image perturbations caused by radio transmission.

Figure 4.5: Camera images from the robot's perspective.

- The robot's rotation causes vertical, wavy distortions of the whole image. Currently, this is dealt with by stopping the robot each time an image analysis is requested.
- Usually, a certain amount of noise is contained in images. A filter operator employing a small Gaussian kernel will take care of this.

4.4 Feature Extraction

The conversion of edge strings to line segments requires a gradient image of the input to be computed. Both, edge detection and line segment extraction algorithms are already implemented in the open source library Gandalf[5], which contains computer vision and numerical algorithms in C and is published under the Lesser GPL license. It turned out to be a handy and fast library regarding fundamental data structures and image processing routines.

The process of feature extraction goes as follows:

1. A new image is requested from the frame-grabber and allocated in RGB colour format of type `unsigned char`.
2. This image is converted to grey level format of type `float`, as required in the following stages.
3. A check for perturbations either rejects or accepts this image in accordance to Equation 4.3.

[5] *Gandalf Home Page*, `http://gandalf-library.sourceforge.net`, [Online], 09/07/2002.

4. The image is convolved in x- and y-direction, with a small Gaussian convolution mask in order to reduce small, thermal noise.

5. The Canny edge detection algorithm [Canny, 1986] is applied to produce a map of edgels and strings.

6. A recursive orthogonal regression line finding algorithm is applied to generate a map of line segments from the edge map.

These operations have several parameters, which have been manually adapted in experiments according to the criteria given in Section 4.2, e.g., to tolerate noise, to focus on distinctive landmarks, to favour long lines, and to take computational costs into account. Figure 4.2 illustrates an example result of the feature extraction process using the set of parameters that was used throughout all experiments.

Among others, the entries of the list of line segments gained comprise the endpoints (vertices) v_1, v_2, and the strength s, a numerical value exhibiting the strength or contrast of the line feature in accordance to the gradient image.

4.5 World Model and Coordinate Frames

Since no 3D world model of our office environment had been available, a new one has been generated by hand. The selection of landmarks to include herein was based on the feature extraction results of the method and its setup given above in order to ease the subsequent matching process.

The format that the model was stored in conforms to the VRML 2.0 standard (cf. Carey and Bell [1997]). This format was chosen because it is very prominent for modelling three-dimensional objects, and can be visualised with the help of a virtual world browser. Another nice feature of the VRML format that has been implemented in our simple parser is modularity — once a node is defined, it can be reused in subsequent nodes.

Please note, that the line segments perceived by the camera and possibly contained in the world model stem from various sources, e.g., changes in texture, shape, structure, illumination or colour.

The following coordinate systems are involved, while each of them is of metric Cartesian type.

World, w: A specific point in the environment was chosen for the origin of the world model. It is right-handed and comprises the three-dimensional coordinates $\mathrm{X} = (X, Y, Z)^{\mathrm{T}}$ with a precision of 1cm.

Robot, r: Apart from that, the robot's global world coordinates are specified by the two-dimensional position of its center, together with the global orientation in the world, $\mathrm{X_r} = (X, Z, \theta)_{\mathrm{r}}^{\mathrm{T}}$, and is subsequently designated robot pose. This infers the assumption that the robot is moving on the ground plane (X, Z) only, but conforms to the common use in the robotics community.

Robot, R: We will also make use of a local, robot centered coordinate system, where locations specify a two-dimensional spatial deviation from the robot's center and a rotational deviation from its facing direction, $\mathrm{x_R} = (x, y, \theta)_{\mathrm{R}}^{\mathrm{T}}$.

Camera, c: Furthermore, the camera centered coordinate system will play an important role subsequently. It is of three-dimensional, right handed Cartesian type and denoted $\mathrm{x_c} = (x, y, z)_{\mathrm{c}}^{\mathrm{T}}$. Being mounted on the robot, the origin is located at a fixed point relative to the robot's center. The camera can be rotated in two degrees of freedom, α, β, and can zoom.

Image, i: The origin of the two-dimensional image coordinate system is positioned at the top left corner of the image, while the camera's main axis goes perpendicular to the image plane through the principal point laying in the image plane. The image properties are determined by parameters of the frame-grabber. Here, a high resolution was chosen (NTSC, 720×480) in order to yield a good accuracy of the geometric features to be detected. Note, that the data volume is reduced drastically as soon as the line segments have been extracted.

World Model Projection

Recalling that the idea of the devised sensor model at hand works by comparing the measurements with corresponding expectations, the camera view needs to be simulated according to a given camera pose in the world yielding line segments in the image plane. Coming from the most basic form of projection in Equations 4.1, translational and rotational transformation between the world and the camera coordinate systems have to be introduced as follows:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix}_i = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathrm{t} \\ 0_3^{\mathrm{T}} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{4.4}$$

where Equations 4.1 are part of the so-called camera calibration matrix $\mathbf{K}$

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Hereof, the vector $\mathbf{t}$ specifies the translation between the world and the camera origin. The rotational transformation matrix is determined by

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where α, β, γ are the Euler angles called pitch, yaw and roll angles according to the "xyz" convention. Note that the directions of angles and the order of the matrices are significant. Using the so-called projection matrix $\mathbf{P}$, the model of linear camera projection is defined by $\mathrm{x}_i = \mathbf{P}\mathrm{X}$.

Clipping the world model

Before the projection can be performed, the given world model has to be clipped to that part visible according to a given camera pose, similarly to the projection of real scenes.

In order to ease the task of computing the part of the world model laying within the viewing volume of the camera, i.e., those features that become projected onto the image plane, the global world coordinates attached to each entry of the world model representation are converted to the camera coordinate system: $\mathrm{x_c} = \mathbf{R}\mathrm{X} + \mathrm{t}$. The viewing volume is defined by four planes originating in the origin of the camera coordinate system, where the corner vectors are

$$
\begin{aligned}
\mathrm{a} &= (-c_x, -c_y, 2f)^{\mathrm{T}} \\
\mathrm{b} &= (-c_x, c_y, 2f)^{\mathrm{T}} \\
\mathrm{c} &= (c_x, c_y, 2f)^{\mathrm{T}} \\
\mathrm{d} &= (c_x, -c_y, 2f)^{\mathrm{T}}
\end{aligned}
$$

Here, c_x, c_y are the width and height of the CCD chip plane, respectively. The algorithm in my implementation for clipping the model edges proceeds as follows for each edge $e = (\mathrm{p}, \mathrm{q})$ comprising the vertices p and q:

1. If $p_z \leq 0$ and $q_z \leq 0$ then discard e, since it is located completely behind the camera ($p = (p_x, p_y, p_z)^{\mathrm{T}}$). This test eliminates the back of the camera in the world model, which would otherwise be projected by Equation 4.4, too.

2. The test whether e is completely located within the viewing volume is performed in the camera centered, spherical coordinate system with the boundaries $(\rho, \pm\alpha_v, \pm\alpha_h)$ given by

$$
\begin{aligned}
\alpha_v &= \arctan(c_y, 2f) \\
\alpha_h &= \arctan(c_x, 2f)
\end{aligned}
$$

   for both, p and q, where the initial ray of the spherical coordinate frame coincides with the positive main camera axis. If this holds, e belongs to the new list of clipped edges and the next edge will be considered in step 1.

3. Next, the intersections of model edge e with the four planes given by $(0, \mathrm{a}, \mathrm{b})$, $(0, \mathrm{b}, \mathrm{c})$, $(0, \mathrm{c}, \mathrm{d})$, and $(0, \mathrm{d}, \mathrm{a})$ are computed, where at most two intersections lay within the boundaries $(\rho, \pm\alpha_v, \pm\alpha_h)$ of the spherical coordinate frame.

4. Finally, these are used to determine the section of e laying within the viewing volume.

The algorithm does not consider occlusions of model edges. But this is not of great importance as long as the structures in individual rooms are quite flat as is the case in the world model at hand for the corridor, which is illustrated in Figure 4.6. Making use of hierarchical definitions of scene nodes in the VRML standard and a hypothesis of the robot's pose, the room that the robot is in can be determined and the corresponding world node fetched from the model package.

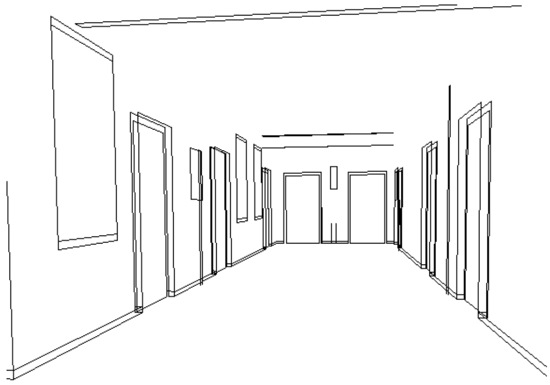


Figure 4.6: The world model of our corridor.

4.6 Matching process

The image feature extraction process described in Section 4.4 and the world model feature extraction of Section 4.5 provide a list of lines in the image plane in geometric form. Now, this section describes how a comparison of both feature maps can be performed resulting in a measure of similarity.

Classification

Matching algorithms can be distinguished by the image model in use.

Raster-based: This class considers the regular pixel-based description of images on the lowest level of data abstraction (signal-level). Here, techniques directly use the amplitude (and colour) values of pixels or functions thereof. The aim is to find the parameters of the transformation between the two given images. Examples of prominent algorithms are Cross Correlation and Least Squares Matching. They are computationally expensive, very specific and therefore often restricted to image patches, where relatively precise matches are needed. While they are very accurate, they are intolerant of different lighting conditions, and especially of occlusions. Common similarity measures include the correlation coefficient and the L^p distance.

Feature-based: This class considers feature attributes on a higher, symbolic abstraction level. Often, features are sufficient for describing the content of an image. While they are computationally less demanding and more invariant to effects like illumination etc., a suitable type of features has to be selected. Since feature extraction represents the interpretion of the image in terms of a certain semantic, it infers a loss in information, which is typically desired in order to focus on certain properties while substantially reducing the amount of data.

Sometimes, the spatial precision of the image is reduced by the abstraction from signal-level to feature-level.

Relations: Optionally, a topology of the image at feature-level can be extracted. Comparing relationships within the topological trees of both images, one can gain robust matching algorithms.

Semantics: Human or machinery classifiers assign interpretations to image segments on the highest abstraction level. This allows to generalise and categorise at the semantic level, e.g., employing semantic networks.

Research on similarity measures at the feature level has been pursued in the field of image retrieval. In opposite to the task at hand, feature vectors are precomputed in image retrieval and stored in a database for the sake of performance.

Requirements to the Similarity Measure

Clearly, the inaccuracy of the camera pose introduces differences between the expected and the measured feature maps. Regarding the intersection of features in the image frame with small pose differences, the difference between both images can usually be modelled by an affine transformation. Measuring real world data, interferences and noise will be present, of course. The similarity measure should be able to cope with additions in the measurements with respect to the world model, which is very important. Besides static objects not present in the world model, dynamic objects may be perceived as well.

Furthermore, the following differences in feature space have been observed to be common when comparing camera images and their corresponding expectations extracted from the world model:

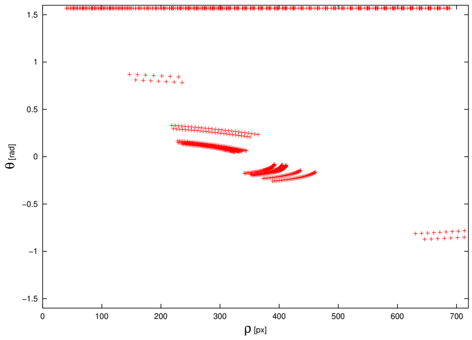
1. Splitting: An image edge is split into several line segments instead of a single one. First, this can be inevitably caused by lighting effects, occlusions by dynamic objects etc. Furthermore, imprecision in the given world model and occlusions by model objects possibly can occur.

2. Duplication: Lighting effects, especially shading can yield duplications of a single real-world edge, which are approximately similar and in parallel to the expected one. But, details not modelled of the real environment like multiple lines of a door frame occur more often.

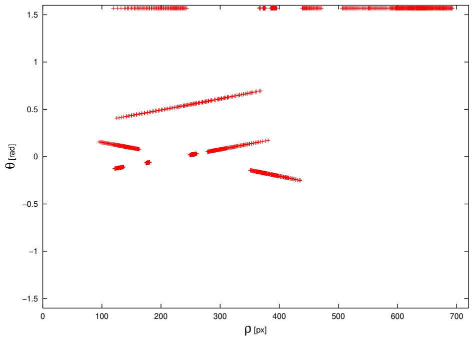
An appropriate method and space of comparison was in question that can deal best with such effects.

The idea in our project is to convert the given set of line segments to "points" in the Hough space, particularly because of two handy properties. First, the representation of lines in the Hough space tolerates "holes". Second, lines of similar orientation and length of its normal through the origin are grouped together. Hence, neglecting the Euclidean position of the endpoints, the desired properties mentioned above are satisfied in this parameter space.

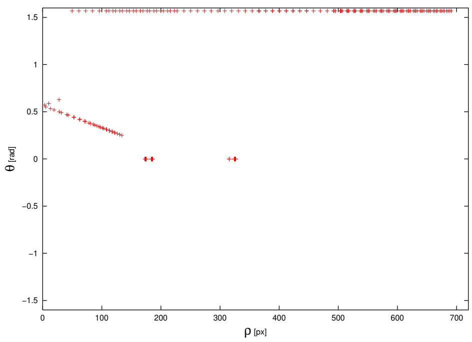
In the Hough space, a given line segment is represented by $\rho = x \cos\theta + y \sin\theta$, where θ denotes the angle with the normal of the line and ρ is the perpendicular distance from the origin. The orientation conforms to the range $\theta \in [0, \pi]$, since the direction of line segments is not significant.



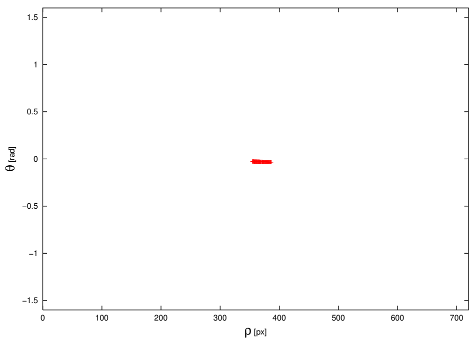
(a) 45 degree turn



(b) 150 cm forwards



(c) 150 cm sideways



(d) 150 cm sideways

Figure 4.7: Camera motion in Hough space

Figure 4.7 illustrates typical traces of world model features during motion in the Hough space obtained from projection of the world model into the image plane and conversion to the Hough space. Hereof, Figure 4.7(a) was obtained from a stepwise 45 degree turn in place with step-size 1 degree. It shows the perspective effect of turning in the world model comprising horizontal and vertical lines only, resulting in an elliptical motion of horizontal lines of the world model and linear motion of vertical lines in the Hough space. Besides turning, moving forward and sideways are typical motions of our robot as depicted in Figures a to c, respectively.

While Figures 4.7(a) to 4.7(c) contain many features making these locations suitable for our sensor model, Figure 4.7(d) was obtained from moving sideways along the corridor for the same distance as in Figure 4.7(b) or Figure 4.7(c), but perceiving very few features. Actually, Figure 4.7(d) depicts a trace of two Hough points representing the corner between the floor and the wall, which only change their positions due to a 10 degree difference in alignment with the wall. Note that this is not a particular weakness of expressiveness of the Hough space, since the same applies for the image frame.

The following similarity measures have been devised, which are based on the representation in the Hough space.

4.6.1 Similarity Measures

In order to accommodate various tolerances described in Section 4.6 like splitting and duplication, similarity measures either consider a very small region centered at the reference feature only, or weigh the resulting measure with respect to a certain distance measure. Subsequently, the following notation is used:

$s = \Phi(A \rightarrow B)$ denotes the similarity measure between the feature map A and feature map B; s for short. It is computed by mapping features from A to the reference window B. Very often, a similarity measure depends on the direction of comparison, i.e., $\Phi(A \rightarrow B)$ yields a different value as $\Phi(B \rightarrow A)$ (cf. Candocia and Adjouadi [1997]).

H_e is the set of **e**xpected Hough points that was generated from the world model, where $h \in H$ denotes a single Hough point;

H_m is the set of **m**easured Hough points obtained from perceived image frame by feature extraction;

$|H|$ denotes the cardinality of the set H;

$\#_{i=0}^{N}(p_i)$ defines a function which takes a logical formula as argument, evaluates the truth value for each predicate index i and returns the number of positive truth values.

If not stated differently, the value of the similarity measures stays within the range $s \in [0, 1]$, but needs to get converted into a probability later.

4.6.1.1 Centered Grid Match Count

The motivation for this similarity measure was to devise a fast and simple measure based on very local matches. Its main idea is to compute the proportion of model features that can be confirmed by the given image measurement. The simple method implemented and investigated is to check whether a given image feature falls into a rectangular range in Hough space centered around the reference point of the given model feature. In the positive case, the reference feature is indicated to be confirmed by the measurement. We defined this similarity measure s by:

$$\begin{aligned} s &= \begin{cases} 0, & \text{if } |H_e| = 0 \\ 1/\,|H_e|\,\#_{i=1}^{|H_e|} \max\left(1, \#_{j=1}^{|H_m|} p(h_{e_i}, h_{m_j})\right), & \text{otherwise} \end{cases} \\ p(h_1, h_2) &\equiv |\rho_1 - \rho_2| < \rho_{\max} \wedge \|\theta_1 - \theta_2\| < \theta_{\max} \end{aligned} \tag{4.5}$$

The predicate p functions as a local, rectangular mask determined by the thresholds $\rho_{\max}$ and $\theta_{\max}$. Note that the maximisation $m(1, \cdot)$ addresses the problem of split lines in measurements in particular: Perceiving a single line within a bounded vicinity of the expected feature already confirms this expected line feature.

Note that the denominator of Equation 4.5 yields normalisation in terms of the number of features present in the feature map generated from the world model. Similarly does the maximum operator. By that, a general similarity measure is gained that is commensurable among all feature maps.

4.6.1.2 Global Grid Match Count

Again, computational efficiency is the main motivation of the following similarity measure, though it has only a slightly different implementation. Here, we define a global grid used to determine the range within which a positive match is counted. While Equation 4.5 remains the same, the predicate p is now defined by:

$$p(h_1, h_2) \equiv (\rho_1 \div \rho_{\text{res}}) = (\rho_2 \div \rho_{\text{res}}) \wedge (\theta_1 \div \theta_{\text{res}}) = (\theta_2 \div \theta_{\text{res}})$$

where $\div$ denotes the integer division and $\rho_{\text{res}} \times \theta_{\text{res}}$ defines the grid resolution.

Recalling the criterion mentioned in Section 4.6 to introduce as little heuristics as possible and to keep the similarity measure simple, these criteria are met by the two given above.

4.6.1.3 Nearest Neighbour

The following similarity measure originates from the field of computational geometry and refers to the Nearest Neighbour (NN) Problem (cf. Smid, 2000). Hereof, the aim is to identify the point from a given set of points that is nearest to a certain reference point according to a measure of distance.

Our implementation adopts the Nearest Neighbour approach to the definition of a geometric similarity measure. It simply searches among the whole set of points and remembers the minimal distance discovered so far for each given reference point. Then, the similarity measure is formed by the average of these distances as follows:

$$s = \begin{cases} 0, & \text{if } |H_e| = 0 \\ 1/|H_e| \sum_{i=1}^{|H_m|} \max_{j \in |H_e|} f_d(h_{m_i}, h_{e_j}), & \text{otherwise} \end{cases} \tag{4.6}$$

$$f_d(h_1, h_2) = \left(1 - \frac{|\rho_1 - \rho_2|}{\rho_{\max}}\right) \left(\frac{\|\theta_1 - \theta_2\|}{\theta_{\max}}\right)^2 \left(1 - \frac{|l_1 - l_2|}{l_{\max}}\right) \tag{4.7}$$

Equation 4.6 defines the straightforward, exact Nearest Neighbour search. While NN is a generic algorithm, it is based on the distance measure. The distance measure in Equation 4.7 combines three independent components: the distance in the ρ-coordinate, the distance in the θ-coordinate and the difference in length of the two given lines. In order to gain a commensurable definition for similarity measures, the components are normalised. Those components considering $\triangle\rho$ and $\triangle l$ employ linear weights, and for the $\triangle\theta$-component I tried a quadratic scheme as a kind of heuristic, which still conforms to the range $s \in [0; 1]$.

In comparison to the previous similarity measures, this one involves a simple heuristic and takes the feature attribute length into account. On one hand, the latter improves distinctiveness of features, but on the other hand by that the similarity measure becomes subject to false negative classifications, because it will often fail matching split lines with their corresponding model lines. Nevertheless, this relationship is investigated using that similarity measure.

4.6.1.4 The Hausdorff Distance

The Hausdorff distance is a prominent distance measure for comparing two sets of geometric entities in the Euclidean space (cf. Federer, 1969). Here, we adopt this distance measure for a definition of a geometric similarity measure for feature maps.

In contrast to correspondence-based matching techniques, this method does not compare pairings of entities, i.e., relationships are not established. In the project at hand, we implemented and investigated the Hausdorff distance using the following definition of the similarity measure s:

$$\begin{aligned} s &= \begin{cases} 0, & \text{if } |H_e| = 0 \text{ or } |H_m| = 0 \\ 1/f_H(H_m, H_e), & \text{otherwise} \end{cases} \\ f_H(H_1, H_2) &= \max\left(1 + \max_i \min_j f_h(h_{1_i}, h_{2_j}), 1 + \max_i \min_j \left(f_h(h_{2_i}, h_{1_j})\right)\right) \\ f_h(h_1, h_2) &= \sqrt{\left(\frac{\rho_1 - \rho_2}{\rho_{\max}}\right)^2 + \left(\frac{\|\theta_1 - \theta_2\|}{\theta_{\max}}\right)^2} \end{aligned} \tag{4.8}$$

Apparently, this similarity measure causes high computational costs, since it involves two nested loops evaluated twice. Please note that the normalisation values, $\rho_{\max}$ and $\theta_{\max}$ in Equation 4.8 are given constants, given by the image size and π, respectively. The $\|\cdot\|$ operator normalises its argument to the angular range of the Hough space.

4.6.1.5 Grid Length Match Proportion

Finally, this similarity measure should restrict the matching range to a local grid cell, and consider all feature attributes. In contrast to the similarity measure based on counting feature matches with discrete grid bounds explicated in Section 4.6.1.2, this time the lengths of the line segments are significant and strongly influence the similarity measure.

The idea is to return the proportion of the length of a certain model line segment that is present in the measured image. In order to allow for some tolerance of the position in the Hough space, the lengths of all measured lines that fall into a certain grid cell in the Hough space are summed up. Likewise, the lengths of all Hough points generated from the world model that fall into the same grid cell are summed up. This similarity measure computing the proportion of matching line lengths based on a global grid is defined as follows:

$$
\begin{aligned}
c &= \#_i^{G_w} \#_j^{G_h} \left(g_{e_{i,j}} > 0 \right) && (4.9)\\
g_{m_{i,j}} &= \sum_k^{|H_m|} p(h_{m_k}, i, j) l_k && (4.10)\\
g_{e_{i,j}} &= \sum_k^{|H_e|} p(h_{e_k}, i, j) l_k && (4.11)\\
s &= \begin{cases} 0, & \text{if } c = 0 \\ 1/c \sum_i^{G_w} \sum_{j, g_{e_{i,j}} > 0}^{G_h} \min\left(1, g_{m_{i,j}} / g_{e_{i,j}}\right), & \text{otherwise} \end{cases} && (4.12)\\
p(h, i, j) &\equiv (\rho \div \rho_{\text{res}} = i) \wedge (\theta \div \theta_{\text{res}} = j)
\end{aligned}
$$

Equation 4.9 defines the number of grid cells c that contain model features in the Hough space. Before this count can be computed by simply iterating over the whole grid of size $G_w \times G_h$, the sum of the lengths of those Hough points, $g_{m_{i,j}}$ and $g_{e_{j,j}}$, that fall into a given grid cell (i, j) for the **m**easured, and the **e**xpected feature map, respectively, is computed. The multiplication of the predicate p with the length of the feature at hand denotes a masking operation similar to the Dirac delta function and restricts the sum of lengths to the bounds of grid cell (i, j).

There are two important remarks to be emphasised here.

1. This definition exhibits a strong direction of similarity, i.e., the difference between $\Phi(A \to B)$ and $\Phi(B \to A)$ could be considerably high. This is introduced on purpose by the predicate in Equation 4.9 and the minimisation in Equation 4.12, because the world model represents the reference frame of both feature maps in the comparison.

2. Using the world model as the reference window (and normalising accordingly), allows for features sensed additionally in the image. By that, the similarity measure should neither considerably suffer from static features that are not modelled, nor from (small) dynamic objects or noise, since these are likely to be discarded in Equation 4.12 by the condition $g_{e_{i,j}} > 0$.

Unfortunately, this similarity measure infers high computational costs in terms of time and memory, because two grids are necessary for storing the sums of lengths, $\{g_{e_{i,j}}\}$ and $\{g_{m_{i,j}}\}$, that have to be set up, filled and finally matched. Nevertheless, it is interesting to experience the influence of the line segment length and the particular method employed here.

The empirical results for these similarity measures are presented subsequently in Chapter 5.

Chapter 5

Experiments and Results

A particular configuration of a sensor model comprises the similarity measure employed among those introduced in Section 4.6.1 and a set of parameters.

Two sets of empirical simulation have been performed for each sensor model. First, extensive computer simulations were used during the initial development and evaluation stage of the sensor model, where the perception of images was simulated in addition to the feature generation step based on the world model. Finally, after the sensor model had been devised and tested, the individual sensor models were obtained from detailled computer simulations described in Section 5.1. Then, real-world data was aquired using the robot in our office environment, which was employed in the determination of the sensor model based on real-world images and is described in Section 5.2.

In both cases, the camera sensor model was obtained by stochastic sampling.

5.1 Computer Simulations on the Sensor Model

In order to evaluate the suitability of the devised similarity measures, many computer simulations have been performed, which are described in this section.

The idea of these simulations is to generate a synthetic feature map for the measurement, another one for the corresponding expectation, but to introduce a difference between both poses beforehand. Then, each particular configuration of the camera sensor model is evaluated in order to yield a function of deviation from the reference pose.

In each experiment, the similarity measure yields the probability $p(m|l_m, l_e, w)$ of measuring (the image/feature vector) m at location l_m given the reference location l_e (the hypothesis or estimate) within the world model w. Since it relates to particular, absolute world locations and a particular scene including illumination conditions, object structures etc., the result, $p(m|l_m, l_e, w)$, is not comparable and hence not of great use. Hence, the approach of this project to achieve a generic camera sensor model is:

1. To abstract from specific locations and conditions: By averaging over a high number of empirical measurements it is assumed that the feature-based percept gains independence from a specific world location.

2. To abstract from specific, absolute world coordinates: A generic, commensurable coordinate frame is obtained by converting the tuple of absolute poses $\langle l_m, l_e \rangle$ into a three-dimensional pose deviation $\triangle l$.

In accordance to these assumptions, $p(m|l_m, l_e, w) = p(m|\triangle l, w)$ holds, which conforms to the description of sensor models given in Section 3.3.

5.1.1 Configurations

For now, both data sets, the measurement and the expectation are simulated and both employ the same world model. Several different setups of the simulation have been implemented:

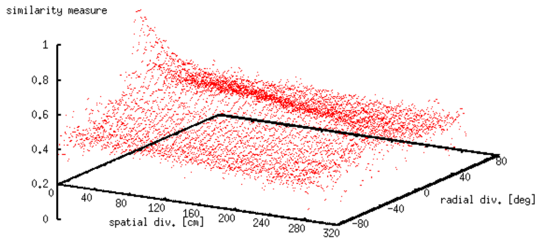
1. The location of the feature map (both, the measurement and the expectation) can be set to a fixed, specified pose.
2. The deviation in a location can be constrained to arbitrary combinations of independent deviations in the individual coordinate axes, x, y and θ, including no deviation at all.
3. The deviation can be simulated in both, the robot's and the world coordinate system.
4. It is possible to generate a random location within the world model for the first, and a random location within certain bounds around the first for the other pose[1].
5. The division of the world model into cells of a uniform 3D grid allows systematic simulation within the whole model.

Choosing from the possibilities 1 to 3, the given similarity measure can be evaluated under particular conditions. Configurations 4 and 5 are well suited for checking the ability of some camera sensor model to generalise its behaviour to many, various poses within the world model. Since the former yield very specific results used in the development stage, those are omitted here for the sake of brevity.

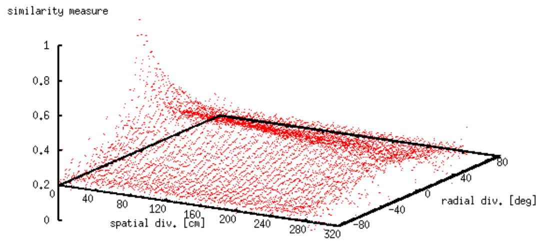
5.1.2 Results of Simulation Using Idealised Conditions

Figure 5.1 depicts the results of the conducted computer simulations of the project at hand. For the sake of illustration, the deviations in the x- and y-coordinate are combined into a single spatial deviation in the Figures 5.1(a) to 5.1(e) employing the L_2 norm. The diagrams show the similarity measure over the spatial and the radial deviation. Combining $\triangle x$ with $\triangle y$ into a single spatial deviation value $||\triangle x + \triangle y||$ should be confirmed by the observation that the behaviour of the similarity measure does not differ significantly considering $\triangle x$ and $\triangle y$ separately. This is depicted in Figure 5.1(f), where separate deviations in single axes are averaged over 1000 random locations.

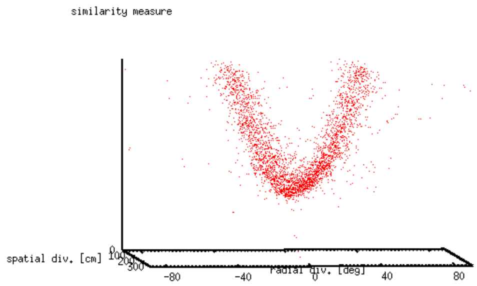
[1] Due to memory constraints and accuracy demands, an unconstrained deviation is difficult to realise. We simulated using resolutions for the 3D grid of up to $(\pm 3cm \times \pm 3cm \times \pm 1°)$ and ranges of up to $(\pm 500cm \times \pm 500cm \times \pm 180°)$ as will be shown subsequently.



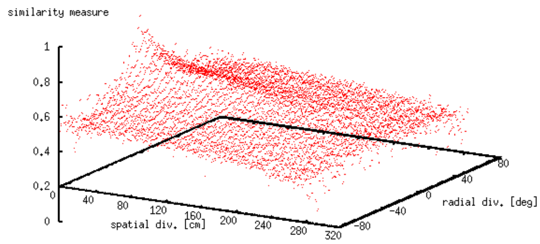
(a) Centered grid match count



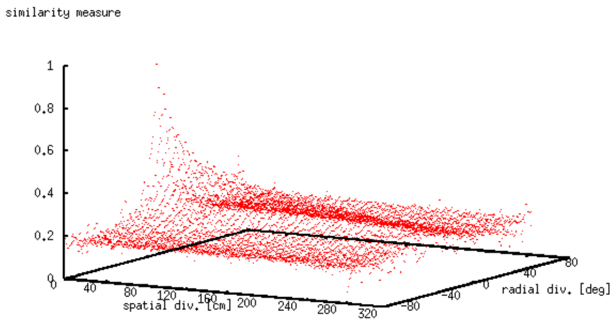
(b) Global grid match count



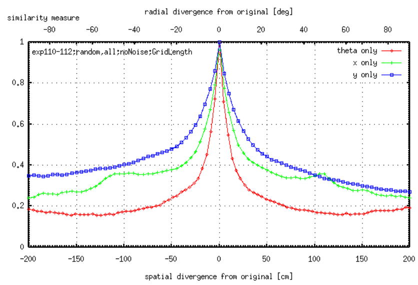
(c) Nearest Neighbour



(d) Hausdorff Distance



(e) Grid length match proportion



(f) Deviations in single axes separately

Figure 5.1: The results of computer simulations of similarity measures

For better illustration, Gnuplot's smoothing filter "dgrid3d" has been applied to the Figures 5.1(a) to 5.1(e), which discretises the input data by a regular, user-specified grid and accumulates each cell's representative c_j by a weighed average of all samples. Each weight $w_{j,i}$ is inversely proportional to the distance between the data sample d_i and the grid point g_j at hand:

$$\begin{aligned} w_{j,i} &= \frac{1}{\|g_j - d_i\|_n}; \quad i \in [1, D],\, j \in [1, G] \\ c_j &= \frac{1}{D} \sum_i^D w_{j,i} d_i \end{aligned}$$

where D specifies the number of data samples given and G the number of grid cells. The grid's resolution (G) as well as the degree of the vector norm, n, are parameters to be provided.

The configuration 4 was used for the simulations in all three axes of deviation in Figures 5.1(a) to 5.1(e), i.e., some location of the measurement was chosen randomly within the world model, and for that a deviation vector within a centered 3D grid around the first location was generated randomly. At least 15 million random pose tuples for each similarity measure have been integrated into a single grid containing the average similarity measure given the corresponding 3D deviation in location. The size of the pose deviation accepted was $(\pm 200cm \times \pm 200cm \times \pm 90°)$ with $(5cm \times 5cm \times 2°)$ being the resolution.

As was explained in Section 3.3, the average over the results of the similarity measure of all runs having the same, discrete deviation vector is taken to yield an approximately general similarity value independent to particular locations and conditions. At least 15 million similarity values have been accumulated for each run of computing samples of the vision sensor model employing a particular similarity measure such that on average circa 20 samples represent a single entry of the model (a dot in the diagrams of Figure 5.1). By that, in the whole over 500 hours computing time were spent to accomplish 124 trials.

The diagrams show considerably different results of the individual similarity measures. Obviously, all except the one using the Nearest Neighbour approach yield a similar shape of the expected type, although the same world model, namely a corridor of width 3.4 m and length 18 m was used. My explanation for the inverse behaviour of the NN similarity measure refers to its strong dependence on the distribution of features in the reference frame: On average, the density of features is higher when looking along the longer axis of the corridor than looking perpendicular to it. Since the correspondence search of the Nearest Neighbour approach is not constrained as it is in the three grid-like based approaches, it is more likely to find a matching feature in the near vicinity of some reference feature simply when the feature density is higher. Since there are two directions of looking along the longer axis of the corridor, it is possible to gain a higher feature density by turning 45 degrees.

The diagrams show that all three grid-like implementations yield results that look similar. There appears to be only a minor difference between the centered grid approach in Figure 5.1(a) and the global grid approach in Figure 5.1(b), as anticipated due to minimal implementational differences. Comparing the grid-based similarity measure of Figure 5.1(b) that considers the 2D position of features in the Hough space "only" with the one of Figure 5.1(e) that considers the length of each feature in addition to (ρ, θ), the latter results in a peak that is a bit more distinctive. This confirms our expectations of taking an additional feature attribute into account for the similarity measure, since the length attribute might be very informative in terms of classification.

5.1.3 Towards Realistic Simulation

Of course, the results given above are not very significant concerning real world applications, since ideal measurements were predicted from the world model. In order to gain an estimate of the behaviour of the devised similarity measures using real-world measurements, a model representing real-world noise and distortion has been implemented (error model for short). This is supposed to simulate several different kinds of distortions and noise. The motivation is to gain information about the tendency of the influence of noise and distortion on the behaviour of the similarity measures. The error model implements the following types of distortions and noise by iteration over all given lines features:

Duplication: Any line segment that has not already been duplicated becomes duplicated with a small probability ($p = 0.2$) and is indicated as such.

Splitting: Any line segment that is of a minimum length ($\geq 30\,\text{px}$) is split with a certain probability ($p = 0.3$). Both new segments are assigned length 40% of the original segment and are indicated being a split copy.

Distortion: A small distortion is applied to the parameters (ρ, θ, l) of each line segment according to zero-mean Gaussian distributions with small deviations.

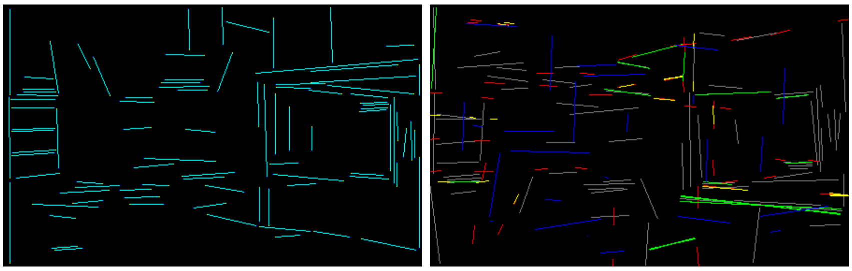
Noise: According to an exponential distribution, a random amount of additional lines is scattered in the valid frame by randomly choosing the first point of the line segment according to a uniform distribution. Next, the length of the line segment is drawn from a random Gaussian distribution. Finally, the endpoint of the line is determined by its orientation that is drawn randomly according to the current distribution of orientation angles of lines employing the Roulette Wheel selection algorithm.

Figure 5.2(a) shows a feature map extracted from a typical camera image in our office environment captured from the robot's point of view. The application of the error model to this frame generates the Figure 5.2(b) where different distortion types are illustrated with different colours.

5.1.4 Results of Simulations Using Realistic Conditions

Now, all similarity measures have been simulated according to the same configuration given in Section 5.1, but this time applying the error model to the (simulated) measurement before the matching is evaluated employing the similarity measure under consideration. Figure 5.3 shows the results of all similarity measures except the Nearest Neighbour approach[2]. For comparison, Figure 5.3(e)

[2] The similarity measure using a Nearest Neighbour approach has been confirmed to yield negative behaviour in these simulation runs, too.



(a) Original feature map (b) Distorted feature map

Figure 5.2: Result of applying the error model.

depicts the simulation of deviation in single coordinate axes separately. In comparison with Figure 5.1(f), the peak of all three curves remains at the zero deviation vector. Although the curves are flatter when noise is introduced, the gradients still have the same direction. Furthermore, the signs of two corresponding gradient maps are the same.

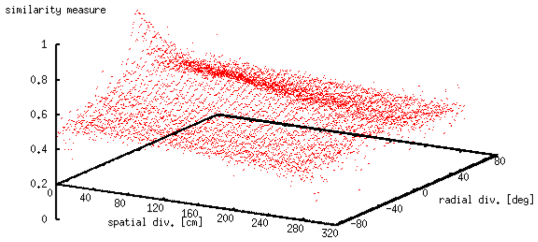
Looking at the diagrams concerning deviations in three axes, all displayed similarity measures again exhibit a ridge along the x-axis at zero radial deviation. The degree of the slope of each ridge and the difference between the minimum and the maximum similarity value changed most compared with the diagrams of ideal simulations.

Comparing the ideal and the distorted diagrams of the Hausdorff Distance approach, the distinctiveness of the peak has significantly deteriorated in the second one. Due to the complexity of the computation, the reason is difficult to discover. Hence, this similarity measure is rather inappropriate in the light of future applications.

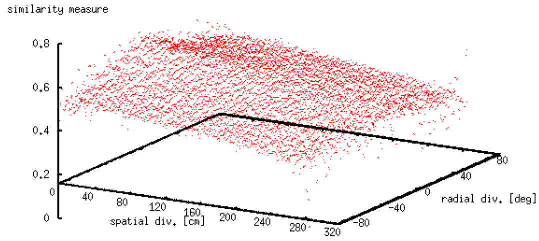
The similarity measure based on a centered grid match count revealed a rather robust behaviour, which is depicted in Figures 5.1(a) and 5.3(a). With the introduction of the error model, the whole surface raised by circa $s = 0.1$, and the peak shrunk accordingly. But the slope of the ridge remained approximately the same. Both diagrams exhibit the feature of a stronger distinctiveness in the axis of orientation, θ, as is alike in all other diagrams of Figure 5.1. Therefore, this version based on the grid match count appears to be well suited for the similarity measure of a camera sensor model.

Due to minor implementation issues, Figures 5.1(a) and 5.1(c) look alike. Again, the whole surface of the Diagram 5.1(c) raised by circa $s = 0.1$ and the peak value shrunk to almost $s = 0.8$. Nevertheless, the similarity function still distinguishes very well between the best matching location at the zero deviation vector and any other location in the diagram; and, the slant of the ridge easily reveals the correct gradient to the peak of the surface.

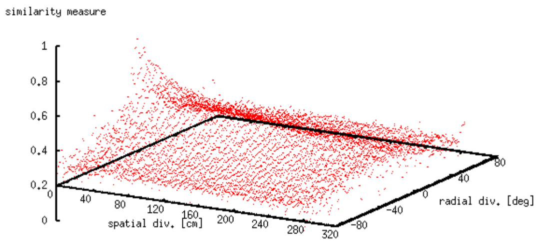
Finally, the error model seems to mainly affect the strength of the surface's peak in Figure 5.3(d) and its near surroundings, while the shape, gradients and the height of the rest remained approximately the same. The peak of Figure 5.3(d) now reaches $s = 0.6$ in comparison to $s = 0.9$ in Figure 5.1(e). In



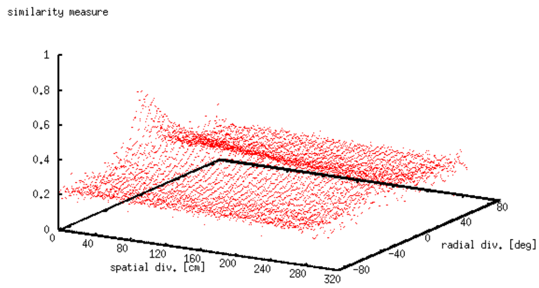
(a) Centered grid match count



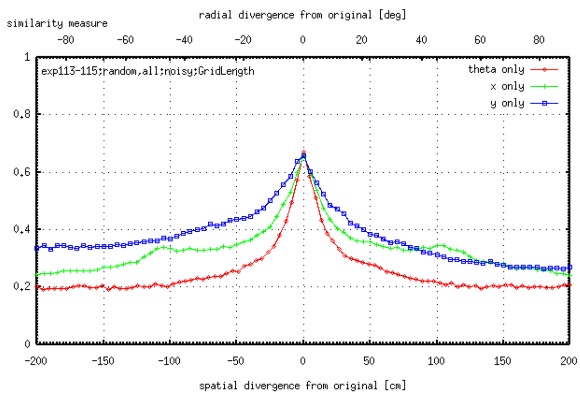
(b) Hausdorff Distance



(c) Global grid match count



(d) Grid length match proportion



(e) Deviations in single axes separately

Figure 5.3: Computer simulations of similarity measures including noise.

contrast to the similarity measures based on the correspondence match count gaining a high length match proportion appears to be less likely, resulting in a difference of $\triangle s = 0.4$ between the bottom and the top of the surface taking the length attribute into account, as compared to $\triangle s = 0.6$ for the simple match count.

This explanation is based on the types of distortions to be expected in real-world data, which is simulated in the error model. First, splitting of line segments is a common distortion resulting in several parts whose summed lengths do not reach the original length. Next, each individual feature is exposed to distortion which likely displaces the segment to a different discrete grid cell.

5.1.5 Conclusions

In summary neglecting the rather minor implementational differences between the centered and the global grid, the computer simulations showed that the following two similarity measures turned out to be suitable for the application in a vision-based sensor model:

1. The measure based on the count of matched correspondences in a global grid;

2. The one based on the proportion of line segment lengths confirmed by the measurement.

It remains to further investigations and real-world experiments, whether taking the attribute length into account is advantageous for the particular application in mind.

Furthermore, it remains questionable whether all assumptions hold (p. 46), all heuristics and parameters used are sound, and the synthetic error model is suitable in our context. Therefore, real-world experiments should be performed, which are described in Section 5.2.

The actual probabilistic camera sensor model is obtained from the value computed by the similarity measure, s, by normalisation over the space of pose deviations $\triangle l$. In particular, the probability of measuring a camera image m given the pose hypothesis l is defined by

$$p(m|l,w) = \frac{1}{\int s(m,\triangle l,w)\mathrm{d}(\triangle l)} s(m,\triangle l,w)$$

given the world model w. This refers to the structure given in Figure 4.1.

5.2 Experiments on Real Robot

Similarly to Section 5.1, where the camera sensor model was computed experiments using a stochastic sampling approach, this section describes the experiments based on real-world data. Now, the acquired data is used to compute the camera sensor model by means of stochastic sampling, but this time based on more realistic conditions.

5.2.1 Configuration and Procedure

The aim of the experiments described subsequently was to collect real-world data with the Sony CCD camera mounted on our Pioneer 2 mobile robot in the corridor environment depicted in Figure 4.6. This should return an estimate on how the sensor model performs in real world and how big the discrepancy between those and the results from synthetic simulations is.

After having collected and registered the measurements, they are fed into the camera sensor model using various configurations. First, an arbitrary position from which a typical scene can be perceived was chosen and measured by hand in order to obtain its world coordinates almost precisely. Starting from here, a simple program steers the robot while the video stream is broadcasted to a frame-grabber as follows:

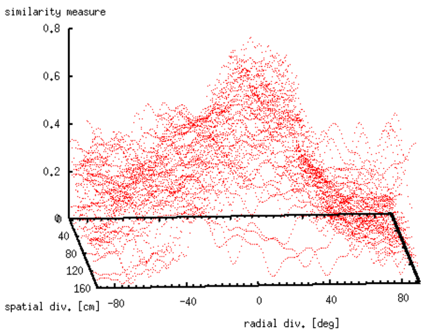
1. While the full turn is not yet completed, turn a little bit (5 or 10 degrees);
   (a) Wait until the robot has completely stopped for avoiding image distortions. Store both, an image from the camera and the corresponding pose in the world;
2. Now, the robot is placed to a different vertex of a regular, virtual grid of discretised deviations from the starting point — this is measured by hand. Return to step 1 until many grid points have been visited.

Being a very time consuming task due to manual measurements and frequent stops in robot movement (see Section 4.3), two different starting points were chosen, around which a total of 3203 valid image-pose tuples have been collected at 64 locations. In order to retain the flexibility of using the measurements in several different configurations and to save storage space, only the line segment features have been extracted from each image at the highest resolution.

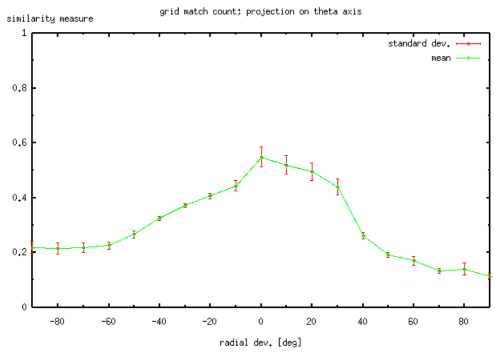
Then, from each set S_i of samples that referred to the same starting point (two in this case), an arbitrary reference pose r_i is chosen. Next, the expected feature map is generated with respect to r_i from the world model w. For each measurement–pose tuple $(m, y) \in S_i$, the camera sensor model is employed to compute the probability $p(m|\triangle l, w)$, where $\triangle l = |r_i - y|$ and the abstraction $p(m|y, r_i, w) = p(m|\triangle l, w)$ is assumed (see page 5.1). Similarly to the computer simulations of Section 5.1.1, the three-dimensional deviation vector $\triangle l$ is discretised and probabilities referring to the same instance of $\triangle l$ are averaged in order to approximate the assumption of independence of particular poses and conditions.

5.2.2 Results and Discussion

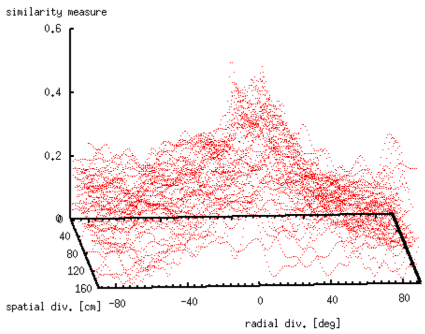
The procedure of applying the camera sensor model to the real world measurements accumulated beforehand has been performed using different configurations concerning the similarity measure and parameters in use. The results are depicted in Figure 5.4. Analogous to Figure 5.1, the left column shows the value of the similarity measure $s(\triangle \mathrm{l}, w)$ given the three-dimensional deviation vector $\triangle \mathrm{l} = (\triangle x, \triangle y, \triangle \theta)^T$ in relative, camera centered coordinates. Again for the sake of illustration, $\triangle x$ and $\triangle y$ are integrated into a single spatial deviation $||\triangle x + \triangle y||$ using the L_2-norm, and a smoothing filter (dgrid3d, see p. 49) was applied.



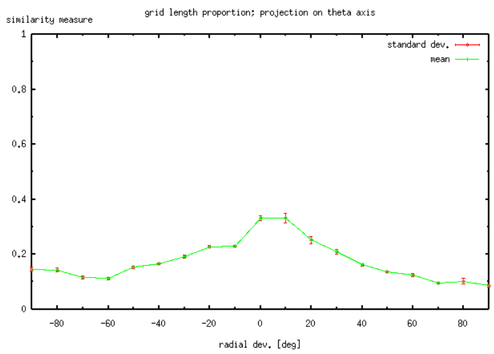
(a) Global grid match count



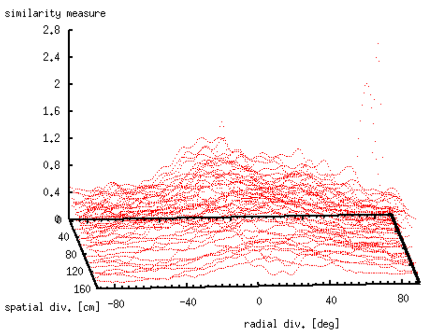
(b) Global grid match count, profile



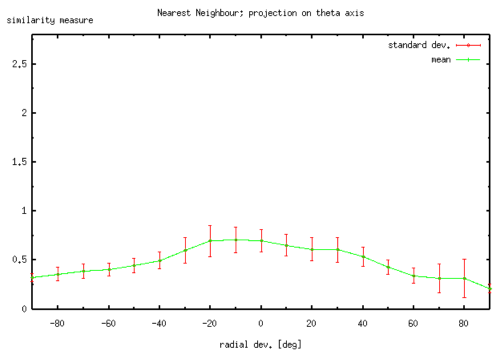
(c) Grid length match proportion



(d) Grid length match proportion, profile



(e) Nearest Neighbour



(f) Nearest Neighbour, profile

Figure 5.4: Results of real-world experiments.

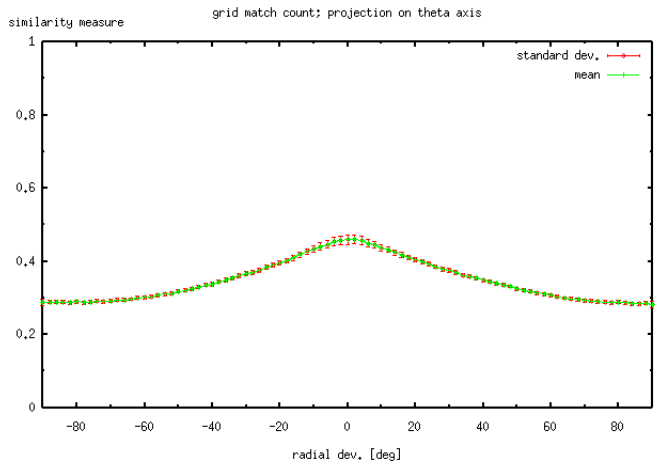


Figure 5.5: Global grid match count, obtained from computer simulations, mean and standard deviation.

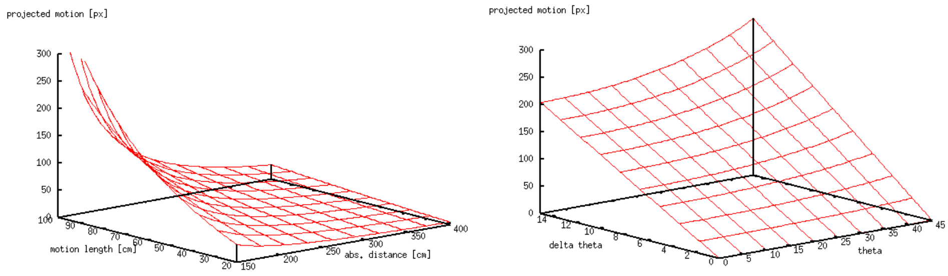
Apparently, the difference in values along the axis of spatial deviation at fixed radial deviation is less than those of the results obtained in computer simulations shown in Figure 5.1. In order to gain a measure of comparison, the diagrams in the right column show the mean value and standard deviation along the axis of spatial deviation at a fixed radial deviation corresponding to each diagram on its left. These diagrams confirm that there exists a single distinctive peak in all three experiments at approximately $\triangle\theta = 0°$.

From Figure 5.4 one can see that the similarity measure "global grid match count" yields a better distinction due to bigger range $[0.1; 0.55]$ than the "grid length match proportion" bearing range $[0.1; 0.35]$. Moreover, the standard deviations are less in the latter case.

Recalling the unexpected, inverse behaviour of the Nearest Neighbour approach shown in Figure 5.1(c), it yielded good results this time in real-world application. Although the peak is not as distinctive as depicted in Figures 5.4(e)–(f) being roughly of the same value for the range $\triangle\theta \in [-20°; 30°]$, it exhibits a high standard deviation. Since the surface is rather smooth, there exist gradients considerably different to zero within the profile, and hence a gradient ascent search would be possible here. This contrasts to the other diagrams, Figures 5.4(b) and 5.4(d), where the interval representing the standard deviation often is negligently low.

For the sake of brevity, no diagrams of the similarity measure "centered grid match count" are shown, because they resemble those of the "global grid match count" due to implementational similarity. Furthermore, no results of the Hausdorff-distance approach are shown, because the whole surface looks rather flat with no distinctive peak, thus is not appropriate.

Summarising these results, the three similarity measures depicted in Figure 5.4 appear to be appropriate in general. While the maximum similarity is easily recognised at zero deviation, $\triangle\theta = 0$, considering radial deviation only, there exists no significant difference in the values of the profile along the axis



(a) Forward motion

(b) Rotation

Figure 5.6: Projection of camera motion to image plane.

of spatial deviation keeping $\triangle\theta$ fixed. The influence of this observation is to be seen in the suitability of the devised camera sensor model for real-world application (according to the empirical data available to date): Using the devised sensor model enables the agent to recognise its angular deviation from the assumed pose (hypothesis) given a model of the environment, but indicates difficulties recognising spatial deviation.

The explanation for this result is to be seen in the camera projection to the image plane. In the given experiments, the camera was mounted on top of our robot at height $h = 52\,cm$ pointing forward with approximately zero pan, tilt and roll angle. While the roll angle is fixed by construction, the other two angles can be controlled. Accordingly for typical robot motion and camera views, the skirting boards are often recognised features. Looking at the effect of camera motion on the feature map in the image plane concerning static objects perceptible as line features on the floor, the following can be seen:

Forward motion: Using the linear pinhole camera model, a camera motion by $\triangle Z$ observing a static object on the floor at distance Z results in a change by $\triangle y$ pixels in vertical image coordinates according to

$$\triangle y = f_y h \frac{\triangle Z}{(Z - \triangle Z)Z}$$

Since f_y and h are constants, the function $\triangle y = g(\triangle Z, Z)$ is quadratic in Z, which is depicted in Figure 5.6(a) as a function of changes in the image plane $\triangle y$ over distance of forward motion $\triangle Z$ while tracking an object at distance Z.

Rotation: The camera rotation (or radial object motion) by $\triangle\theta$ at a relative angle of θ approximated by the linear camera model results in a change in horizontal pixel coordinates $\triangle x$ according to

$$\triangle x = \frac{2f_x}{\cos\theta} \tan\left(\frac{\triangle\theta}{2}\right)$$

This exhibits the two-dimensional projection with independence to Z loosing depth information. The relationship, $\triangle x = g(\triangle\theta, \theta)$, is illustrated in Figure 5.6(b) being almost linear in $\triangle\theta$ for small values.

For instance, while a forward motion by $\triangle Z = 10\,cm$ observing an object on the floor in distance $Z = 250\,cm$ results in a change by circa 5 pixels, a rotation by $1°$ with relative angle to the observed object $\theta = 0$ yields a translation by circa 15 pixels in the image plane. *Since the effect of forward motion decreases exponentially with distance to the object, that is why the probability density function of the camera sensor model obtained in experiments shows little dependence on spatial deviation.*

For a comparison between real-world experiments and computer simulations, Figure 5.5 depicts the two-dimensional profile of mean values and standard deviations of the sensor model obtained in computer simulations concerning fixed angular deviation along the axis of spatial deviation. Here one can see, that the deviation intervals are very small as well, similarly to those of Figure 5.4. Hence, the real-world experiments confirm the findings of the simulations with respect to better recognition of angular deviation. The explanation given above shows that this is not a weakness of the camera sensor model, but is caused by the camera projection properties according to the configuration in use. See Section 6.1 for proposals on improvements.

Chapter 6

Conclusions

The natural complexity and variety of uncertainty present in real-world environments pointed up in the first part of this report should have shown that employing sensor fusion can be an appropriate approach to handle uncertain information.

In the project at hand, the implementation of concepts of sensor fusion was addressed. According to the three stages of sensor fusion — correlation, association and combination — a sensor model is required for each sensor device in the first step, which precisely describes how the environment produces sensory measurements. Besides sonar sensors and a laser range finder, our mobile robot can perceive with a colour CCD camera. While models for the first two sensors are well investigated to date, it turns out to be rather difficult to devise a generic, visual model for the camera.

This project presents a simple and robust sensor model for a CCD camera, which is based on a three-dimensional world model. First, line segment features are extracted from perceived images. Next, a synthetic feature map is generated from the world model given a position estimate of the agent's current pose coordinates. In classical model matching approaches, a correspondence analysis follows next, which is of exponential complexity in the size of the feature maps. In contrast, various measures for an immediate comparison between the measured and the expected features in the two-dimensional image plane have been developed.

A further significant difference between typical approaches to a camera sensor model and the implementation at hand is that the uncertainties in camera location and feature recognition are not propagated in order to constrain the image feature extraction to areas of interest. Here, feature extraction is always applied to the whole image, which consumes the biggest portion of computational costs. Nevertheless, it remains questionable, whether region constraints for feature extraction can yield significant improvements given a large-scale, feature-rich office environment. Instead of explicitly computing the projection of uncertainty in the camera position onto the image plane, here we obtained the probability distribution function for the whole sensor model from stochastic sampling. One one hand, the sensor model has been generated in comprehensive, synthetic computer simulations, where the perception process was imitated inversely including a synthetic error model. On the other hand, the sensor-model has been computed using systematic real-world measurements. By that, all uncertainties involved during the measurement, data alignment steps etc., as well as thermal noise are implicitly represented and obtained from stochastic sampling.

Herein, the abstraction from absolute world poses to relative pose deviations was made, which should be justified by the great amount of samples simulated. The sensor model computed for the camera is represented by a grid-based discretisation of the probability density function, such that its application reduces to a fast, simple lookup.

Various similarity measures have been devised and evaluated extensively. The idea common to all is to perform the feature matching in the Hough space, since it allows for automatic grouping of split line segments — a typical perturbation in camera images. While grid-based and Nearest Neighbour approaches, and the Hausdorff distance have been investigated concerning correspondence constraints, the deviations of line segments in Hough space and optionally the segment's length have been employed concerning the distance measure.

In comparison to similarity measures of the classical approach of vision-based navigation (see page 20), we have devised similarity measures in this project, which do not require the computationally expensive correspondence establishment among features of the measured and the expected feature map. Although being based on simple local matching algorithms, this kind of matching of two feature maps has not been encountered in the literature to date. On the other hand, our project bears the disadvantage in comparison to the classical approach of applying a standard feature extraction algorithm to the whole image, which deteriorates its computational efficiency considerably.

In summary, the basic grid-based similarity measure performed well in terms of recognition of deviation in camera orientation, but shows insufficient recognition of spatial deviation. Real-world experiments mainly confirmed the findings from computer simulations.

Since the devised camera sensor is based on line features, it should be applicable in regular indoor environment comprising of polygonal shapes like our corridor. The choice of line segments as the primitives to use in the matching process provides a robust adaptation to different illumination conditions. Furthermore, being based on line features poses no further restrictions on the combination with different sensor types. The camera sensor model of this project seems to be particularly suitable for sensor fusion with a laser range finder. The probability density represented by the camera sensor model does not constrain the sensor model to the use in a Bayesian framework only — instead, it can be interpreted as an function of evidential support in Dempster-Shafer theory or as a membership function in Fuzzy theory, for instance. Nevertheless, its suitability remains to be shown.

6.1 Future Work

This project devised a camera sensor model for the camera of our robot. Apparently, implementing sensor fusion of perceptive information obtained from the camera and from a distance sensor of the robot, be it sonar sensors or the laser range finder, still needs to be accomplished. The first part of this thesis pointed at prominent techniques with which sensor fusion can be realised.

Major issues in the concept of the sensor model's implementation are flexibility and modularity, such that, e.g., the similarity measure, the specific camera model and the feature extraction procedure could be easily replaced.

Currently, feature extraction from camera image amounts to almost the whole computing demands, while the computational costs required for comparing the perceived feature map with the model-based features is negligible. It may be worthwhile to follow the classical approach to model-based visual navigation as was outlined in Section 3.1, where explicit uncertainties in the camera's position are projected to the image plane in order to constrain feature extraction to regions of interest. Of course, this allows for a certain amount of tolerance only depending on the current confidence.

The author believes that the amount of features that can be recognised in images influences the performance of the matching process with the world model. But to date, the relationship has not been investigated. Although the sensor model being independent to the number of features recognised from the mathematical point of view, a reduced number of features will most likely degrade its quality and robustness.

According to the empirical results and the explanation thereof given in Section 5.2.2, the current configuration of the camera facing forwards seems to be inappropriate. The reason is that forward movement results in a very little movement of projected image lines, which declines quadratically with distance. A very promising investigation is to point the camera sideways, since there are many features and both, forward translation as well as rotation will cause considerable changes in the image plane. Adding code of camera interaction already suffices to adapt the implementation to different camera viewing directions. An even more general solution is active vision, where the camera can be fully controlled on demand, for instance with the aim to always point the camera at areas in the environment known to have desired properties.

Many approaches to visual navigation track the location of distinctive world objects in subsequent images and maintain tracks thereof in the image plane over time. By that, semantics can be propagated to the next frame, which can improve the robustness of navigation. If considering each image independently, much information about temporal changes is lost.

As noted above, most classical approaches propagate uncertainties in order to restrict feature extraction to predicted regions of interest.

If camera images are to be fused with a planar distance sensor, a commensurable coordinate frame and data abstraction level are on demand in order to accomplish data alignment (see Section 2.3). The most obvious solution is to extend the feature extraction process to obtain vertical edges in images and corner features in laser range data.

Appendix A

Statement

- The report has been prepared according to [The University of Chicago, 1982].
- The references have been prepared conforming to [American Psychological Association, 1994].
- The report was typeset using $\LaTeX\ 2_\epsilon$.

Erklärung

Hiermit versichere ich, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Bibliography

Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proc. of the Symposium on Close-Range Photogrammetry*, pages 1–19, Falls Church, VA, 1971. American Society of Photogrammetry.

American Psychological Association. *Publication Manual of the American Psychological Association.* American Psychological Association, Washington, DC, 1994.

S. Atiya and G. D. Hager. Real-time vision-based robot localization. *IEEE Trans. on Robotics and Automation*, 9(6):785–800, 1993.

R. T. Azuma. *Predictive Tracking for Augmented Reality.* PhD thesis, University of North Carolina at Chapel Hill, Dept. of Computer Science, Feb. 1995. Technical report TR95-007.

J. R. Beveridge. *Local Search Algorithms for Geometric Object Recognition: Optimal Correspondence and Pose.* PhD thesis, University of Massachusetts, Amherst, MA, June 1993.

J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

J. Borenstein and Y. Koren. Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Journal of Robotics and Automation*, 7(4):535–539, 1991.

T. G. R. Bower. The evolution of sensory systems. In R. B. MacLeod and H. L. Pick, Jr., editors, *Perception: Essays in Honour of James J. Gibson*, pages 141–153, Ithaca, NY, 1974. Cornell University.

R. A. Brooks. Visual map making for a mobile robot. In *Proc. of the 2nd Int'l IEEE Conf. on Robotics and Automation (ICRA'85; Mar.; St. Louis, MO)*, pages 824–829, 1985.

B. G. Buchanan and E. H. Shortliffe. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, Reading, MA, 1984.

W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proc. of the 15th Nat'l Conf. on Artificial Intelligence (AAAI'98; Madison, WI).* AAAI, 1998a.

W. Burgard, A. Derr, D. Fox, and A. Cremers. Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'98; 13–17 Oct.; Victoria, Canada)*, 1998b.

W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the 13th Nat'l Conf. on Artificial Intelligence (AAAI'96; Aug.; Menlo Park, CA)*. AAAI, 1996.

A. Cameron and H.-L. Wu. Identifying and localizing electrical components: A case study of adaptive goal-directed sensing. In *Proc. of the IEEE Int'l Symposium on Intelligent Control (13-15 Aug.; Arlington, VA)*, pages 495–500. IEEE, 1991.

A. Cameron and H.-L. Wu. Using adaptive goal-directed sensing to overcome sensor uncertainty. In P. S. Schenker, editor, *SPIE Proc. Vol. 1611; Sensor Fusion IV: Control Paradigms and Data Structures*, pages 202–213, Apr. 1992.

F. Candocia and M. Adjouadi. A similarity measure for stereo feature matching. In *IEEE Trans. on Image Processing*, volume 6, pages 1460–1464, Oct. 1997.

J. F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov. 1986.

R. Carey and G. Bell. *The Annotated VRML 2.0 Reference Manual*. Addison-Wesley Longman, Reading, MA, Jan. 1997.

J. A. Castellanos, J. Neira, and J. D. Tardós. Multisensor fusion for simultaneous localization and map building. *IEEE Trans. on Robotics and Automation*, 17 (6):908–914, Dec. 2001.

R. Chatila and J.-P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. of the 2nd Int'l IEEE Conf. on Robotics and Automation (ICRA'85; Mar.; St. Louis, MO)*, 1985.

M. R. B. Clarke. Discussions to 'Belief Functions'. In P. Smets, E. H. Mamdani, D. Dubois, and H. Prade, editors, *Non-Standard Logics for Automated Reasoning*. Academic, London, 1988.

R. T. Collins and J. R. Beveridge. Matching perspective views of coplanar structures using projective unwarping and similarity matching. *Computer Vision and Pattern Recognition*, pages 240–245, June 1993.

I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a multiple hypothesis approach. *Journal of Artificial Intelligence*, 66(2):311–344, 1994.

J. L. Crowley. Navigation for an intelligent mobile robot. *IEEE Trans. on Robotics and Automation*, RA-1(1):31–41, Mar. 1985.

J. L. Crowley. Mathematical foundations of navigation and perception for an autonomous mobile robot. In *Proc. of the Int'l Workshop on Reasoning With Uncertainty in Robotics (RUR'95; 4–6 Dec; Amsterdam, The Netherlands)*, pages 9–51. Springer, 1995.

N. Daucher, M. Dhome, J. T. Lapreste, and G. Rives. Modelled object pose estimation and tracking by monocular vision. In *British Machine Vision Conf.*, volume 1, pages 249–258, 1993.

F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the Condensation Algorithm for robust, vision-based mobile robot localization. In *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'99; June; Fort Collins, CO)*, NY, 1999. IEEE.

A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society*, 30:205–247, 1968. Series B.

DSTO. *Data fusion lexicon.* Department of Defence, Australia, Sept. 1994. Defence Science and Technology Organization (DSTO), Data Fusion Special Interest Group.

L. Du, G. D. Sullivan, and K. D. Baker. Quantitative analysis of the viewpoint consistency constraint in model-based vision. In *Proc. of the 4th Int'l IEEE Conf. on Computer Vision (Berlin; May 1993)*, pages 632–639. IEEE Computer Society, 1993.

H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *The Int'l Journal of Robotics Research*, 6(3):3–24, 1987.

W. Eckstein and C. Steger. Architecture for computer vision application development within the HORUS system. *Journal of Electronic Imaging*, 6(2): 244–261, April 1997.

S. Enderle, M. Ritter, D. Fox, S. Sablatnög, G. Kraetzschmar, and G. Palm. Soccer-robot localization using sporadic visual features. In *Proc. of the 6th Int'l Conf. on Intelligent Autonomous Systems (IAS-6; July; Venice, Italy)*, 2000.

J. E. Evans, T. J. Dasey, D. A. Rhoda, R. E. Cole, F. W. Wilson, and E. R. Williams. Weather sensing and data fusion to improve safety and reduce delays at major west coast airports. Technical report, Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA, Sept. 1999.

P. Fearnhead. *Sequential Monte Carlo methods in filter theory.* PhD thesis, University of Oxford, Oxford, UK, 1998.

H. Federer. *Geometric Measure Theory.* Springer, New York, 1969.

A. J. Fennelly, J. K. Woosley, D. M. McMahon, S. Bhuminder, and J. W. Wolfsberger. Multivariate data spaces and multivariable systems analysis for explosives detection systems using X rays. In *Proc. of the SPIE, X-Ray Detector Physics and Applications (23–24 July 1992; San Diego, CA)*, volume 1736, pages 159–170. SPIE, Feb. 1993.

D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

D. Fox, S. Thrun, W. Burgard, and F. Dellaert. Particle filters for mobile robot localization. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

M. Friedmann, T. Starner, and A. Pentland. Device synchronization using an optimal filter. In *Proc. of the 1992 Symposium on Interactive 3D Graphics (Cambridge, MA)*, pages 57–62, 1992.

R. Gens, Z. Vekerdy, and C. Pohl. Image and data fusion - concept and implementation of a multimedia tutorial. In *Proc. of the 2nd Int'l Conf. on Fusion of Earth Data (26–31 Jan. 1998; Cote d'Azur, France)*, pages 217–222, 1998.

N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. on Radar and Signal Processing*, 140:107–113, 1993.

D. O. Gorodnichy. On using regression in range data fusion. In *Canadian Conf. on Electrical and Computer Engineering (CCECE'99; May 9-12, Edmonton, Canada)*, 1999.

J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'98)*. IEEE/RSJ, 1998.

J.-S. Gutmann, W. Hatzack, I. Herrmann, B. Nebel, F. Rittinger, A. Topor, T. Weigel, and B. Welsch. The CS Freiburg robotic soccer team: Reliable self-localization, multirobot sensor integration, and basic soccer skills. In M. Asada, editor, *RoboCup-98: Robot Soccer World Cup II*, Berlin, 1999. Springer.

J. M. Hammersley and D. Handscomb. *Monte Carlo Methods*. Methuen, London, 1964.

J. E. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatics*, 6:555–563, 1970.

J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9:547–559, 1969.

R. Hanek and T. Schmitt. Vision-based localization and data fusion in a system of cooperating mobile robots. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS'00)*, pages 1199–1204. IEEE/RSJ, 2000.

A. Hauck and N. O. Stöffler. Video-based determination of the joint states of articulated objects. In *Proc. 2nd Int'l Conf. on Robotics, Vision, and Parallel Processing for Industrial Automation (ROVPIA'96; 28–30 Nov.; Ipoh, Malaysia)*, pages 1018–1023, 1996.

M. Herman and T. Kanade. Incremental reconstruction of 3-D scenes from multiple, complex images. *Artificial Intelligence*, 30(3):289–341, Dec. 1986.

D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *Int'l Journal of Computer Vision*, 5(2):195–212, 1990.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic, 1973.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–46, Mar. 1960.

R. E. Kalman and R. Bucy. New results in linear filtering and prediction theory. *Journal of Basic Engineering, Transactions ASME Series D*, 83:95–108, 1961.

K. Konolige and K. Chou. Markov localization using correlation. In *Proc. of the 16th Int'l Joint Conf. on Artificial Intelligence (IJCAI'99; July; Stockholm, Sweden)*, pages 1154–1159. Morgan Kaufmann, 1999.

D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the 12th Nat'l Conf. on Artificial Intelligence (AAAI'94; July; Seattle, WA)*, pages 979–984, 1994.

A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using modelbased reasoning and prediction of uncertainties. *Computer Vision, Graphics, and Image Processing–Image Understanding, (CVGIP'92)*, 56(3): 271–329, Nov. 1992.

P. Krause and D. Clark. *Representing Uncertain Knowledge: An Artificial Intelligence Approach.* Intellect, Oxford, England, 1993.

R. Kumar and A. R. Hanson. Robust methods for pose determination. In *NSF/ARPA Workshop on Performance versus Methodology in Computer Vision*, pages 41–57, Seattle, WA, 1994. University of Washington.

P. Lamb, F. Horowitz, and H. W. Schmidt. A data model for geoscientific spatial information systems. Technical Report TR-HJ-94-05, CSIRO Information Technology, Apr. 1994.

S. Lanser. *Modellbasierte Lokalisation gestützt auf monokulare Videobilder.* PhD thesis, Technische Universität München, Dec. 1997.

J. Liang, C. Shaw, and M. Green. On temporal-spatial realism in the virtual reality environment. *Proc. ACM UIST'91 4th Annual ACM Symposium on User Interface Software and Technology*, pages 19–25, 1991.

D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

D. G. Lowe. Review of "TINA: The Sheffield AIVRU vision system" by J. Porrill et al. In O. Khatib, J. Craig, and T. Lozano-Pérez, editors, *The Robotics Review I*, pages 195–198. MIT, Cambridge, MA, 1989.

L. E. Marks. *The Unity of Senses: Interrelations among the Modalities.* Academic, New York, 1978.

J.-A. Meyer and A. Guillot. From SAB90 to SAB94: Four years of animat research. In D. Cliff, P. Husbands, J.-A. Meyer, and S. W. Wilson, editors, *From Animals to Animats 3. Proc. of the 3rd Int'l Conf. on Simulation of Adaptive Behavior*, Cambridge, MA, 1994. MIT / Bradford Books.

H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.

H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA'85; Mar.; St. Louis, MO)*, pages 116–121, 1985.

R. R. Murphy. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Trans. on Systems, Man, and Cybernetics*, 26(1):42–51, 1996.

G. Oriolo, G. Ulivi, and M. Vendittelli. On-line map building and navigation for autonomous mobile robots. In *IEEE Int'l Conf. on Robotics and Automation (ICRA'95; Nagoya, Japan)*, pages 2900–2906, 1995.

A. Papoulis. Brownian movement and Markoff processes. In *Probability, Random Variables, and Stochastic Processes*, pages 515–553. McGraw-Hill, New York, 2nd edition, 1984.

M. Ribo and A. Pinz. A comparison of three uncertainty calculi for building sonar-based occupancy grids. *Robotics and Autonomous Systems*, 35(3–4): 201–209, 2001.

D. B. Rubin. Using the SIR algorithm to simulate posterior distributions. In J. M. Bernado, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, editors, *Bayesian Statistics 3*, pages 395–402. Oxford University, Oxford, UK, 1988.

S. J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach.* Prentice-Hall, Englewood Cliffs, NJ, 1995.

G. Shafer. *A Mathematical Theory of Evidence.* Princeton University, Princeton, NJ, 1976.

E. H. Shortliffe. *Computer-Based Medical Consultations: MYCIN.* Elsevier, New York, 1976.

R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the Int'l Joint Conf. on Artificial Intelligence (IJCAI'95; Aug.; Montreal, Canada)*, pages 1080–1087, 1995.

M. Smid. Closest-point problems in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 877–935. North-Holland, Amsterdam, Netherlands, 2000.

H. W. Sorenson. Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68, July 1970.

K. Sugihara. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing–Image Understanding, (CVGIP'92)*, 42:112–129, 1988.

The University of Chicago. *The Chicago Manual of Style.* The University of Chicago, Chicago, IL, 14th edition, 1982. A. Diller (editor).

S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.

O. Trullier, S. I. Wiener, A. Berthoz, and J.-A. Meyer. Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544, 1997.

M. W. Watson. Univariate detrending methods with stochastic trends. *Journal of Monetary Economics*, 18:49–75, 1986.

P. Wunsch. *Modellbasierte 3-D Objektlageschätzung für visuell geregelte Greifvorgänge in der Robotik.* PhD thesis, Technische Universität München, Nov. 1997.